

COORDINATION DRAFT 18 November 1996

FINAL DRAFT 31 October 1996

**United States Imagery and Geospatial System (USIGS)
Common Interoperability Interface Facilities (CIIF)/Defense
Information Infrastructure Common Operating Environment
(DII COE) Integration Analysis**

November 18, 1996

Produced in support of the Common Imagery Interoperability Working Group (CIIWG) of the Imagery Standards Management Committee (ISMC).

This is a coordination draft report for review and comment. Please send comments to Howard Markham at MITRE. Email is preferred, at howardm@mitre.org. Phone is welcome, at (703) 883-5731. Marked up copies also OK, Mail Stop Z267, 1820 Dolly Madison Boulevard, McLean, VA 22102.

Executive Summary

Purpose

The purpose of this report is to identify issues associated with implementing the Common Imagery Interoperability Facilities (CIIF) of the United States Imagery and Geospatial System (USIGS) as services of the Defense Information Infrastructure (DII) Common Operating Environment (COE) and as software facilities that use DII COE services. The analysis results in proposed steps that will help ensure good integration of CIIF and DII COE architectures, functions, and data.

Scope

The analysis encompasses the Image Access Services CIIF defined to date (Catalog Access, Image Access, Profile and Notification, Imagery Dissemination) and those services identified in the USIGS architecture for addressing imagery exploitation and a range of support functions such as workflow management and system administration. It includes all aspects of the DII COE that might provide infrastructure support to the CIIF, and all portions of the DII COE that might duplicate or overlap functions of the CIIF. Key issues include the distributed computing services available in the DII COE, DII COE policies for data management, and DII COE rules for packaging and integrating software components. The means by which CIIF services and interfaces could be adopted into the DII COE using the policies and procedures developed by Department of Defense (DoD) Information System Agency (DISA) for that purpose are described.

Background

A number of Department of Defense (DoD) documents define aspects of information system architectures for use by agencies responsible for acquiring and maintaining information systems. Major examples are the Technical Architecture Framework for Information Management (TAFIM), Joint Technical Architecture (JTA), and the DoD Intelligence Information System (DoDIIS) Profile and Technical Reference Model. These documents promote enhanced interoperability across services, organizations, and command echelons through the use of common interfaces defined in open standards specifications.

The DII COE is an integrated software infrastructure that will eventually conform to the JTA, and is intended to be the common software platform used by all DoD Command, Control, Communications, Computers, and Intelligence (C4I) mission applications. The architecture of the DII COE is client-broker-server, making use of the Distributed Computing Environment (DCE) in the near term and adding the Common Object Request Broker Architecture (CORBA) in the longer term. Technical definition and management of the DII COE are the responsibilities of DISA.

The USIGS is intended to encompass both DoD and non-DoD imagery systems. Major elements of the USIGS will consist primarily of DoD resources and technology, and many USIGS services and applications will be required in many DoD organizations. The CIIF defines interoperable interfaces for the USIGS. Technical definition and management of the USIGS and CIIF are the responsibilities of NIMA. The USIGS and DII COE approaches to interoperable interfaces need to be mutually understood for compatibility.

Interoperability Analysis

CIIF integration into the DII COE will entail CIIF APIs becoming DII COE APIs, and CIIF service implementations calling upon DII COE services using the appropriate DII COE APIs. In this study, CIIF services were compared with DII COE services to identify areas of potential overlap. Significant overlap may exist with DII COE function groupings called mapping, charting, geodesy, and imagery (MCG&I), which currently take the form of the Joint Mapping Tool Kit (JMTK), and with data management.

The major technical difference between the CIIF and the DII COE is the approach to interface implementation. The CIIF is an interface specification apart from any implementation. CIIF interfaces are defined in International Standards Organization (ISO) Interface Definition Language (IDL). ISO IDL has a formal clarity and precision and is able to reflect the hierarchical structure of the CIIF service architecture. The DII COE is an implementation whose APIs are determined by the particular software products used to provide services.

As a way of analyzing relationships between the CIIF and the DII COE, their reference models were combined into a new reference model representing both perspectives. The overall structure is that of the POSIX open systems environment reference model. The combined model was further rationalized into the model shown in Figure Ex-1, which has the following features:

- Applications are grouped into categories like those in the CIIF reference model and the Object Management Group (OMG) Object Management Architecture (OMA)
- The DII COE distinction between product-based kernel services and other infrastructure services has been replaced with infrastructure services only and without product references
- The external environment is represented as in the POSIX model rather than as a network and a set of databases
- Data are shown to the side as a separate architectural element that spans the horizontal layers of function

Figure Ex-2 expands the MCG&I and distributed object categories to identify specific services explicitly included in the CIIF Reference Model.

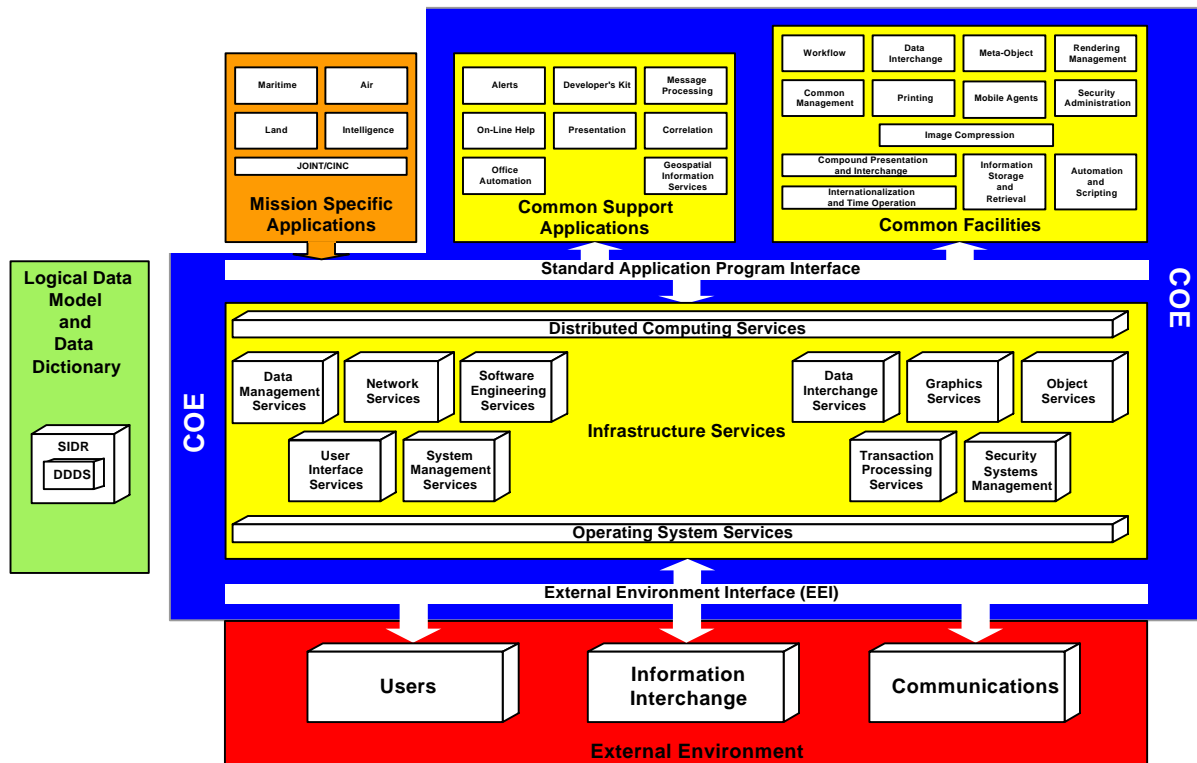


Figure Ex-1. Rationalized DII COE Reference Model

In the spirit of Secretary of Defense William Perry's memorandum of June 1994, "Specifications and Standards—A New Way of Doing Business", both the CIIF and the DII COE efforts require the support of industry, measured by their willingness to produce commercial products using the desired APIs. Enlightened adoption of commercial standards where possible, combined with proactive engagement with industry where Government needs are not being met should be the hallmarks of CIIF and DII COE development and integration.

Conclusions

- The CIIF architecture is compatible with the planned DII COE client-broker-server architecture and the guidance provided by the JTA.
- Integration of the CIIF into the DII COE will depend on the availability of a number of supporting services, including data management, security, system management, and broker-based distributed computing.

- Currently, DII COE services are only partially integrated with the underlying distributed infrastructure. It is not clear when DII COE services will be fully available through a common distributed computing infrastructure (DCE or CORBA).
- There appears to be overlap between the MCG&I services of the DII COE and the services of the CIIF.
- There appears to be overlap between the CIIF Image Access and Catalog Access services and the data management facilities of the DII COE

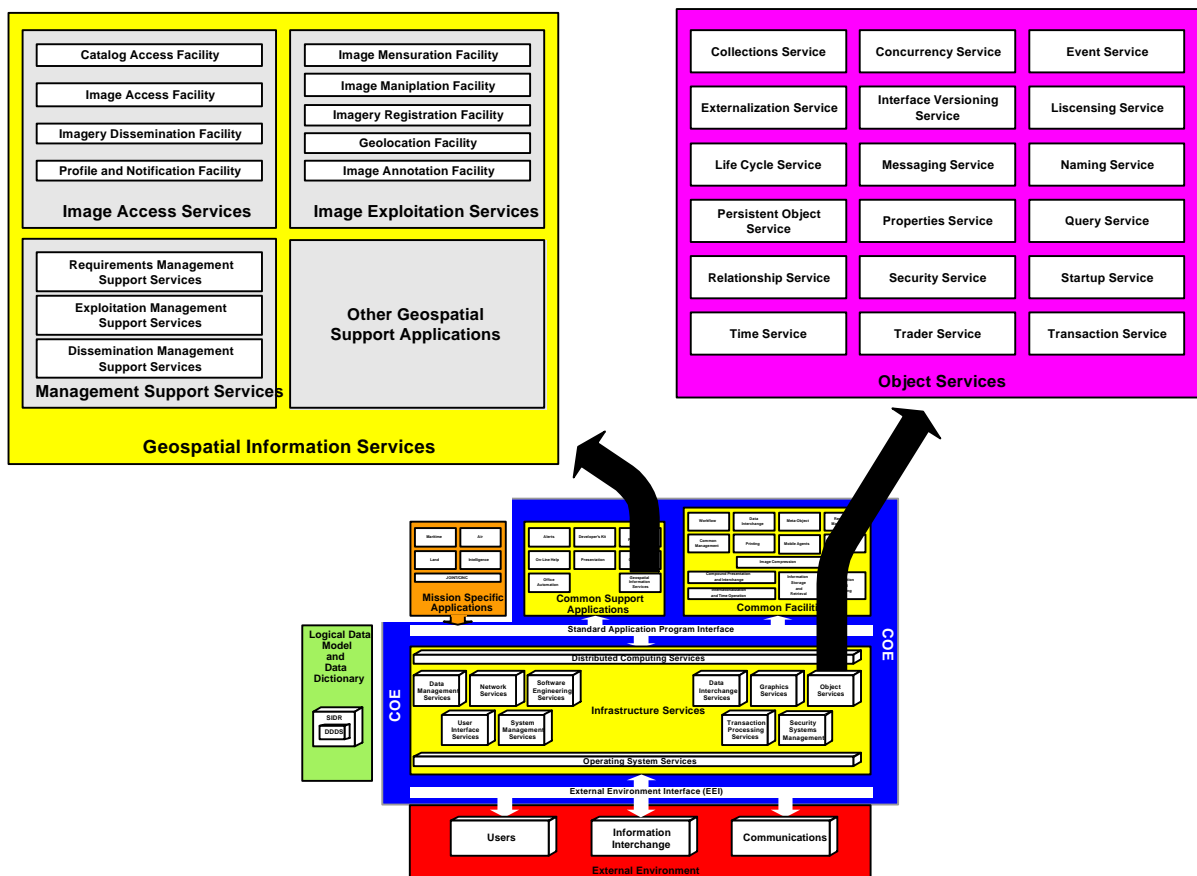


Figure Ex-2. CIIF and Distributed Object Services in the DII COE

Action Plan

To further the integration of CIIF services with the DII COE, the following actions are proposed for the near- and mid-term.

Proposed Actions for NIMA

- Update the CIIF reference model to reflect the findings of this report.

Proposed Actions for DISA

- Adopt the merged reference model of Figure Ex-1 for the DII COE as a more useful portrayal of distributed object services, common facilities, and standardized interfaces.
- Mandate ISO IDL (ISO/IEC DIS 14750) as the preferred language for defining DII COE interfaces.

Proposed Actions for NIMA and DISA

- Promote standards-based rather than product-based APIs.
- Formalize co-participation between the DII COE Architecture Oversight Group and the Imagery Systems Management Committee (ISMC) and their respective working groups.
- Analyze relations between CIIF services and DII COE data management services and plan how they should evolve; extend to other services as needed.
- Add DII COE APIs for common facilities, common support applications, and infrastructure services to the TAFIM and JTA.
- Participate in an ongoing, active Government partnership with industry in standards development and adoption in products.

Preface

This report was prepared during the summer and fall of 1996 by a subgroup of the Common Imagery Interoperability Working Group (CIIWG) led by Henry Rothkopf and Ron Burns. Participants and contributors to the report include the following:

Brad Bretzin, Booz•Allen Hamilton
Ron Burns, NIMA/SEIT
Sam Chang, NIMA/SEEE
Gerry Cookson, Booz•Allen Hamilton
Sheila Daggs, Booz•Allen Hamilton
Ed Hughes, NIMA
Don Joder, Booz•Allen Hamilton
Huet Landry, DISA
Howard Markham, MITRE
Matthew McDermott, Booz•Allen Hamilton
Rick Nehrboss, Booz•Allen Hamilton
Linh Nguyen, TASC
Henry Rothkopf, NIMA/SRMM
John Sarkesain, DISA
Noah Spivak, Booz•Allen Hamilton
Shel Sutton, MITRE

Table of Contents

Section	Page
1 Introduction	1
1.1 Purpose	1
1.2 Background	1
1.3 Scope	2
1.4 Document Structure	3
1.5 Applicable Documents	4
2 Baseline	5
2.1 DII COE	5
2.1.1 Origins in GCCS	6
2.1.2 Objective of the DII COE	6
2.1.3 Services and APIs	8
2.1.4 Phased Implementation	14
2.2 Joint Mapping Toolkit (JMTK)	14
2.2.1 JMTK Program	14
2.2.2 JMTK Description	16
2.2.3 JMTK Schedule	17
2.3 Military Intelligence Database (MIDB) and Image Products Archive (IPA) for GCCS (MIG)	17
2.4 USIGS Architecture and CIIF Reference Model	17
2.4.1 Objective of CIIF	18
2.4.2 Services and APIs	20
2.4.3 Interfaces that Comprise a Facility	26
2.4.4 Phasing of CIIF and IDL Development Activities	27
2.5 Standards Profiles and Technical Architectures	28
2.5.1 TAFIM Reference Model and Standards Profile	28
2.5.2 Joint Technical Architecture (JTA)	29
2.5.3 DoD Intelligence Information System (DoDIIS)	30
2.5.4 Intelink Standards Profile	32
2.5.5 USIGS Standards and Guidelines	33
2.5.6 Imagery Standards Management Committee (ISMC)	34
3 Interoperability Analysis	35
3.1 Overview of Relationship between USIGS CIIF and DII COE	35
3.2 Reference Model Analysis	36
3.3 Architecture Analysis	39
3.4 Interface Analysis	41
3.5 Function Analysis	42

Section	Page
3.6 Compatibility with DII COE Data Architecture and Standards	45
3.7 Packaging for Integration into the DII	48
3.8 DII COE Architecture and Technology Processes	48
3.8.1 The Architecture Oversight Group (AOG) and Technical Working Groups	48
3.8.2 Integration and Runtime Specifications (I&RTS)	49
3.8.3 Analysis and Observations	49
4 Issues and Opportunities	51
4.1 Technology Trends	51
4.1.1 Brokered Distributed Architectures: CORBA, DCE, and DCOM	51
4.1.2 Data Management	52
4.1.3 Web Browsers	52
4.1.4 Mobile Code	52
4.1.5 Security	53
4.2 Reference Models	54
4.3 APIs	54
4.4 Standards Compliance	56
4.5 Data Management	56
4.6 Distributed Computing	57
4.7 Mapping, Charting, Geodesy, and Imagery (MCG&I)	58
4.8 DoD Acquisition and Standards Reform	58
4.9 Conclusions	58
4.10 Action Plan	59
4.10.1 Proposed Actions for NIMA	59
4.10.2 Proposed Actions for DISA	60
4.10.3 Proposed Actions for NIMA and DISA	60
List of References	61
Bibliography	63
Appendix A GCCS and COE Software Segments (Version 3.0)	65
Appendix B DII COE Distributed Computing Primer	69
Appendix C CORBA Requirements in DII COE Distributed Computing SRS	71
Acronyms	75

List of Figures

Figure	Page
1-1 Architectural Context of CIIF	3
2-1 DII as Platform for Mission Applications (Source: I&RTS)	7
2-2 DII COE Reference Model (Source: DISA)	9
2-3 DII COE Schedule of Releases	14
2-4 JMTK Schedule	17
2-5 MIG Schedule	18
2-6 Intelligence Community Reference Model (DRAFT)	19
2-7 USIS Architecture—Digital Elements	20
2-8 CIIF Reference Model	21
2-9 Schedule of CIIF Development Activities	27
2-10 DoD Information Management Integration Model (TAFIM V2.0 Vol. 1)	29
2-11 Detailed DoD Technical Reference Model (TAFIM V2.0 Vol. 2)	31
2-12 DoDIIS Technical Reference Model	33
3-1 Merging of Intelligence Community Reference Model and CIIF Reference Model into DII COE Reference Model	37
3-2 Intelligence and CIIF Reference Models Merged with DII COE Reference Model	38
3-3 Detailed View of MCG&I and Object Management Services in the Merged DII COE Reference Model	40
4-1 Rationalized DII COE Reference Model	55

List of Tables

Table	Page
2-1 DII COE 3.0 Components	12
2-2 COE Compliance Categories	13
2-3 COE Levels of Runtime Compliance	15
3-1 CIIF Common Facilities Mapped to DII COE	44
3-1 CIIF Common Facilities Mapped to DII COE (concluded)	45
3-2 CIIF Imagery Services Mapped to DII COE MCG&I Services	46
3-2 CIIF Imagery Services Mapped to DII COE MCG&I Services (concluded)	47

Section 1

Introduction

The United States Imagery and Geospatial System (USIGS)¹ Architecture [1] addresses the automated collection, storage, retrieval, processing, analysis, and dissemination of national imagery and geospatial data. Within the USIGS Architecture, the Common Imagery Interoperability Facilities (CIIF) define interfaces for imagery services. The imagery community must ensure that systems based on the USIGS architecture can take advantage of infrastructure services being developed by Department of Defense (DoD) organizations and can interoperate with systems in other domains. Particular infrastructures that should be examined in this light include the Defense Information Infrastructure (DII) Common Operating Environment (COE) [2,3] and the DoD Intelligence Information System (DoDIIS) [4,5]. Since the DoDIIS is expected to become aligned with the DII COE, the principal need is to understand the relations between the USIGS and the DII COE.

1.1 Purpose

The purpose of this report is to identify issues associated with implementing the CIIF of the USIGS as services of the DII COE and as software facilities that use DII COE services. The analysis results in proposed steps that will help ensure good integration of CIIF and DII COE architectures, functions, and data.

1.2 Background

DoD information systems are required to conform with the Technical Architecture Framework for Information Management (TAFIM) [6,7]. TAFIM guidance includes a technical reference model based on the Institute of Electrical and Electronic Engineers (IEEE) POSIX Open Systems Environment reference model [8], and a set of information technology standards. The Joint Technical Architecture (JTA) [9] applies the TAFIM for Command, Control, Communications, Computers, and Intelligence (C4I) information systems. The DII COE is regarded as a computing infrastructure that will conform² to the JTA.

¹ The formation of the National Imagery and Mapping Agency (NIMA) on 1 October 1996 has resulted in the expansion of the United States Imagery System (USIS) into the United States Imagery and Geospatial System (USIGS).

² The current DII COE, which includes substantial amounts of legacy code, does not completely conform to the JTA; it will be brought into conformance through incremental upgrades over the next 2-3 years, according to current planning.

Technical definition and management of the DII COE are the responsibilities of the DoD Information Systems Agency (DISA). The DII COE is an outgrowth of the Global Command and Control System (GCCS), initiated in 1993 to create a system that would replace the World-Wide Military Command and Control System (WWMCCS). After a COE was defined for the GCCS, it was perceived as an idea that should be extended to all DoD information systems. The military departments and other mission organizations are expected to design their applications to operate over the DII COE, and to keep DISA informed of infrastructure functions required by the applications. Early versions of the DII COE are made up principally of legacy resources, and include a number of proprietary APIs. However, the goal architecture for the DII COE is full compliance with TAFIM interface standards. It is expected to take several years to migrate the DII COE to the goal architecture.

The USIGS is intended to encompass both DoD and non-DoD imagery systems. Therefore, the USIGS is not formally constrained to conform exclusively to the TAFIM or the JTA. Nevertheless, major elements of the USIGS will consist primarily of DoD resources and technology. In addition, most USIGS services and applications will be required in many DoD organizations. Hence, there are good reasons that the USIGS and DII COE approaches to interoperable interfaces should be mutually understood and compatible. Figure 1-1 illustrates some of the key guidance documents that determine the technology environment in which CIIF services will be deployed.

The CIIF defines interoperable interfaces for imagery functions in the USIGS. Since the interfaces for DoD information services are required to consist of standard APIs specified in the TAFIM, the ability of the USIGS to interoperate with the DII COE is highly dependent on the degree to which CIIF implementations use DII COE APIs to obtain DII services. Conversely, the ability of DoD applications to use CIIF services requires the adoption of CIIF interfaces into the DII COE API suite.

1.3 Scope

The analysis in this report encompasses the Image Access Services CIIF defined to date (Catalog Access, Image Access, Profile and Notification, Imagery Dissemination) and those services identified in the USIGS architecture that address imagery exploitation and a range of support functions such as workflow management and system administration. It includes all aspects of the DII COE that might provide infrastructure support to the CIIF, and all portions of the DII COE, such as data management and mapping, charting, geodesy, and imaging (MCG&I), that might duplicate or overlap functions of the CIIF. Key issues include the distributed computing services available in the DII COE, DII COE policies for data management, and DII COE rules for packaging and integrating software components. The means by which CIIF services and interfaces could be adopted into the DII COE using the policies and procedures developed by DISA for that purpose are described.

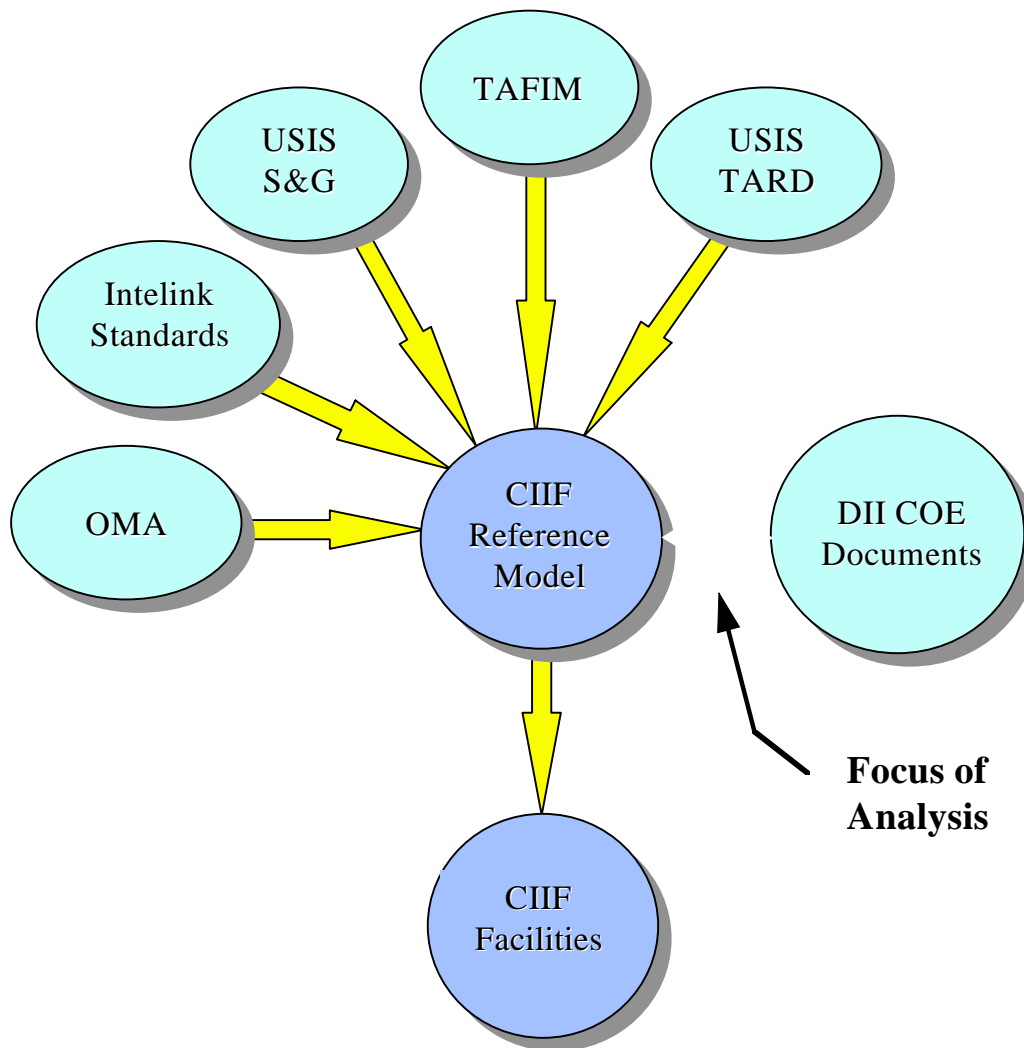


Figure 1-1. Architectural Context of CIIF

1.4 Document Structure

Section 2 summarizes the principal architectures and standards profiles prescribed by DoD for information systems. The DII COE is described at greater length, as is the USIGS, including the CIIF architecture being developed by the national imagery community. Section 3 identifies and analyzes key issues that affect integration of the CIIF and DII COE architectures. Section 4 contains a schedule of proposed actions designed to help ensure that the CIIF will integrate successfully with the DII COE.

1.5 Applicable Documents

This technical report is provided in support of USIGS planning efforts. The USIS and Central Imagery Office (CIO) documents listed in the References section were used directly in the analyses that are documented in this report.

Section 2

Baseline

This section presents overviews of the principal architectures, systems, and programs that affect the compatibility and deployability of the CIIF over the DII COE.

2.1 DII COE

The idea of a Common Operating Environment comes from the Global Command and Control System. The Defense Information Infrastructure COE is an extension from GCCS to all DoD mission applications. The initial implementation of the DII COE is the GCCS COE. The efforts of the past year to define a DII COE will begin to appear late this year when DII COE Version 3 is deployed.

Several key documents characterize the DII COE, among them the following:

- **DII Master Plan (Executive Summary)**, November 6, 1995 [2]. A doctrinal view of the role of the DII COE, its principal components, and the DII COE responsibilities of DoD organizations. The goal is a stable application environment based on open standards and proven components, capable of evolving through component upgrades and replacements, and scalable to the needs of each site.
- **Architectural Design Document for the Global Command and Control System (GCCS) Common Operating Environment (COE)**, December 15, 1995 [3]. High-level requirements and design principles. A secure, open, transparent, flexible, scalable, distributed, fault tolerant infrastructure having a client/broker/server organization is described.
- **DII COE Integration and Runtime Specification (I&RTS), Version 2.0**, October 23, 1995 [10]. A description of how software intended for use in a DII environment must be packaged and tested, to include DII COE compliance checklists. Also describes developer aids and tools available from DISA.
- **DII COE System Requirements Specification (SRS), draft**, July 1996 [11]. Working draft oriented to Version 4.0 of the DII COE. Platform Services, Common Support Applications, and Software Development Services. Being developed in thirteen parts (see below).
- **GCCS Version 3.0 DCE Implementation Plan**, September 29, 1995 [12]. Describes staged approach to implementing DCE: Stage 1 (COE 2.0, in progress) for developers and administrators; stage 2 (COE 3.0), deploy to field and begin legacy conversions. A division of the DII into DCE cells is described.

- **[GCCS] CORBA Migration Strategy Document, draft**, September 25, 1995 [13]. A high level, preliminary discussion. Proposes DCE-based ORBs for the DII COE.
- **DII COE Version 2.0 Series Baseline Specifications**, June 28, 1996 [14]. Identifies the software components of the current DII COE.

2.1.1 Origins in GCCS

The GCCS has two main objectives: the replacement of the World-Wide Military Command and Control System (WWMCCS) and the implementation of the C4I for the Warrior concept. The GCCS will provide a single view of the military C4I for the joint warfighter. The view will be through a widely distributed, user-driven network to which the fighter “plugs in.” The GCCS consists of command and control applications for maritime, air, Joint/CINC, land, and intelligence organizations operating over a common software infrastructure, called the COE. The COE includes both support applications and platform services. The GCCS COE architecture initially defined services in 19 functional areas. In the DII COE, these have been combined and restructured into approximately 14 service areas.

2.1.2 Objective of the DII COE

The DII COE is a major component of the global “info-sphere” and is driven by the C4I For the Warrior vision to provide a fused, real-time, true representation of the three-dimensional battlespace and the ability to coordinate in all directions. The goals of the DII COE are to promote interoperability among applications, access to data independent of its location, and information processing systems and methods that are reliable and scaleable.

The DII COE is described as a software architecture, an approach for building interoperable systems, a common collection of reusable software components, a software infrastructure, and a set of guidelines and standards. It includes the ideas that domain implementations provide their services in the form of standard modularized software that is consistent with the TAFIM technical reference model, and that application programmers have access to these services through standard APIs. The DII COE may be adapted and tailored to meet the specific requirements of a domain.

The DII COE is a comprehensive infrastructure over which most DoD mission applications will operate. This idea is shown in Figure 2-1. The objective architecture is a client/broker/server architecture: clients request services through a broker; servers register services with a broker; brokers find servers for clients. A three-tier structure is envisioned, consisting of user workstations, application servers, and data servers. DII COE components comply with TAFIM standards as much as possible and are migrating to full compliance. Non-developmental items—both commercial off-the-shelf and government off-the-shelf—are preferred over custom components. It is intended that the DII COE will operate on a range of open systems platforms having standards-based operating systems.

The DII COE architecture seeks to establish a more stable, economical, and interoperable infrastructure for mission applications through two key steps:

- Specification of standard APIs, so that applications are portable to any part of the DII, and are isolated from changes to service implementations
- Engineering of a shared set of service implementations, so that duplication of development and maintenance can be reduced to desirable levels.

By isolating applications from changes to the infrastructure, standard APIs for infrastructure services give the infrastructure engineering team greater freedom to improve the infrastructure incrementally to keep pace with technology. In the case of the DII COE, which required a number of compromises on integration in the interests of a rapid initial deployment using legacy implementations, one line of improvement made possible by the freedom to change the service infrastructure is a more thorough integration of components of the infrastructure.

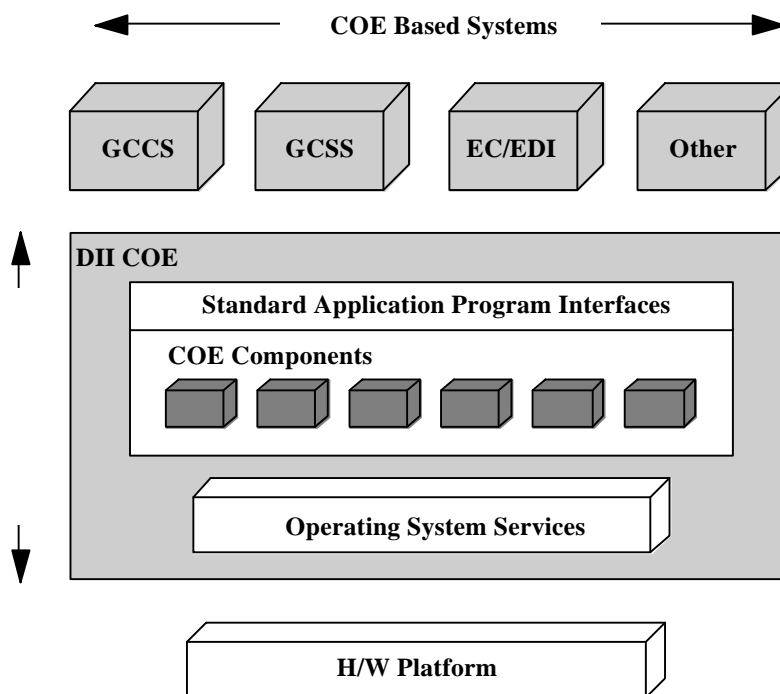


Figure 2-1. DII as Platform for Mission Applications (Source: I&RTS)

2.1.3 Services and APIs

The DII COE taxonomy includes Common Support Applications and Infrastructure Services. Common Support Applications offer services that may be widely used by other applications. Infrastructure Services, also called Platform Services, are software services necessary to move data through the network. Figure 2-2 is an elaboration of the DII COE to show specific service areas. The number of service areas, their names, and whether they are platform services or support applications has been an area of flux in DII definition documents. Figure 2-2 is believed to be the current view and is the basis of discussion in this document.

The service areas are briefly described below. Much of the description has been excerpted from the draft System Requirements Specifications being developed for each area.

Platform Services

- **Communications Services.** An infrastructure of coordinated services primarily supporting connectivity and data exchange between one mission application system or workstation and another. Communications services provide the capability to send, receive, forward, and manage electronic and voice messages. They also provide real-time information exchange services in support of interpersonal conferences. These services include Personal Message Transfer, Organizational Message Transfer, enhanced telephony, shared screen, teleconferencing, and broadcast.
- **Data Management.** Includes File Access, File Management, Database Access, and Database Management. Includes definition, storage, and retrieval of files, databases, and object bases distributed over the network. Includes data exchange facilities between users, computers, and databases.
- **Distributed Computing.** Capabilities that permit procedures and objects to be invoked on remote hosts as though they were local to the calling module. In addition to these basic capabilities, the distributed computing component will include a variety of enabling services, such as security, time, persistence, and naming; many of these services are required for the development of applications that are distributed. The two fundamental technologies that will be implemented in the COE are the Distributed Computing Environment (DCE) and the Common Object Request Broker Architecture (CORBA), including some related services.
- **Management Services.** Ability to manage all hardware and software resources in a heterogeneous, distributed information system. Includes network administration, system administration, security administration. Includes the five System Management Functional Areas defined by the International Organization for Standardization (ISO): configuration management, fault management, performance management, security management, and accounting management. Three levels of management have been defined for DII: global, campus, and site.

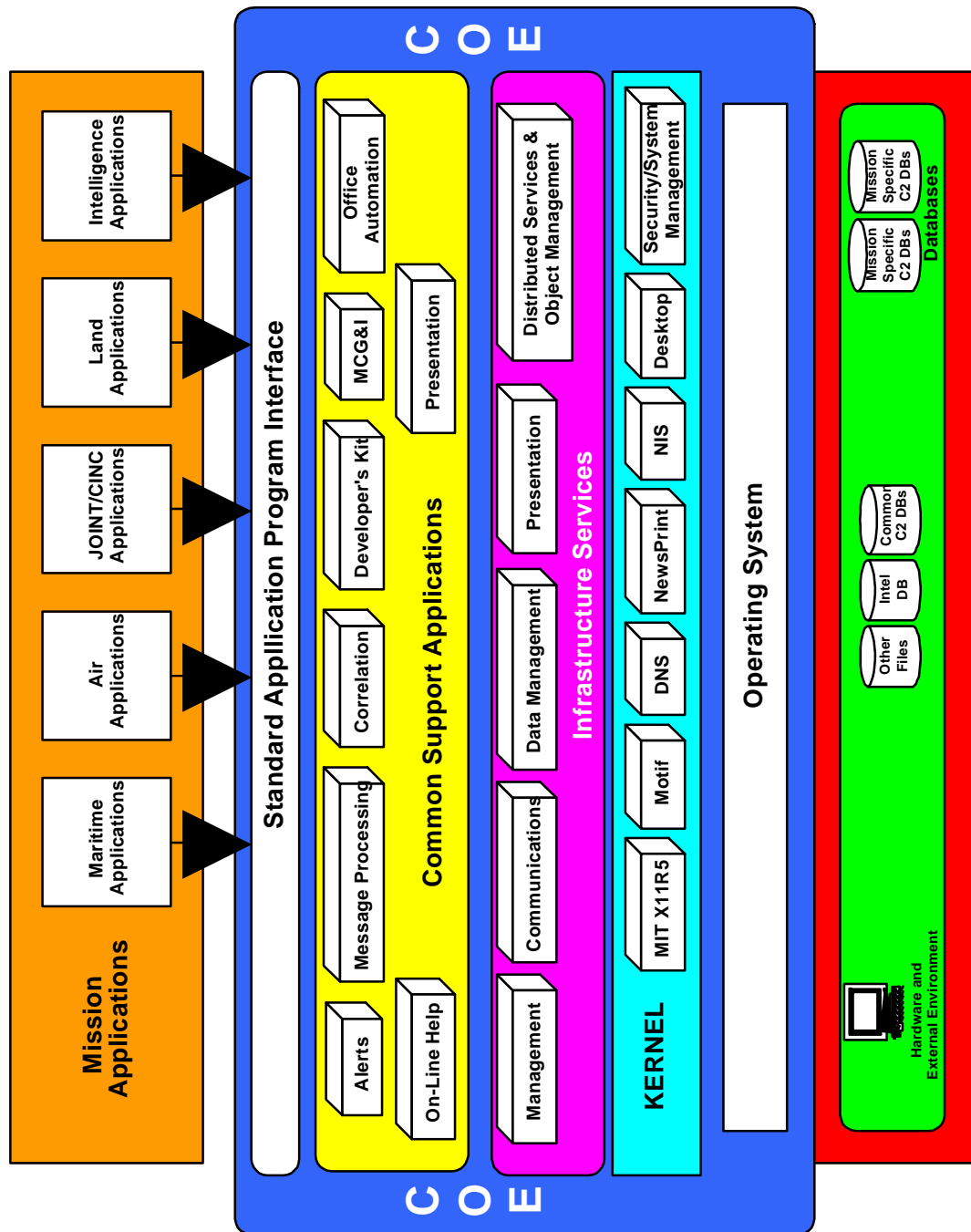


Figure 2-2. DII COE Reference Model (Source: DISA)

- **Presentation Services.** Services required to manage processes and the graphical user interface. Also the software tools required to manipulate and manage multimedia information. These services are defined in two categories—Executive Manager and Multimedia. The executive manager provides process management services for both batch and transaction processing, management of information flow between applications, and notification of critical events. Process management includes information processing, job and process control, menu executive services, security services, and queuing services. All information processing features should be available to Ada and C programs. Multimedia services provide the capability to manipulate and manage information consisting of coordinated text, graphics, audio, imagery, animations and/or video. Data interchange facilities must accommodate specified data formats for all types of media.
- **Security Services.** Five areas of service: Accountability, Access Control, Confidentiality, Integrity, Non-repudiation, and Availability. The DII will initially support a System High mode of operation. The objective of the DII is to support a Multi-level Secure mode of operation, which will demand additional security requirements above those required for System High operation.

Common Support Applications

- **Alert Services.** Generic mechanisms for alerting process. When a process has determined that a predefined criterion or event for notifying other processes has occurred, that process shall use the Alert Services software to notify all interested processes of the event. Alert Services software is composed of an Alerts Server mechanism and a generic Alerts Display mechanism. The Alerts Server allows processes to register to create and receive alerts. The server distributes Alerts using the COE communications support and ensures that Alerts when issued are delivered.
- **Correlation Services.** Maintain a track data base in near real time to include the following information: high level tactical objects to represent platforms (ships, submarines and aircraft), installations both fixed and land mobile, land force units, and technical collection and reporting domain track objects to represent ELINT, COMINT, ACINT, and TADIL tracks.
- **MCG&I Services.** Standard mapping, charting, geodesy, and imagery (MCG&I) data and exploitation capabilities. The MCG&I requirements for the GCCS/JMTK are geospatial analysis, display, spatial database management, and processing of local imagery.
- **Message Processing Services.** Modularized and callable software that supports message parsing, message storage and retrieval, scanning of inbound messages for satisfaction of Standing Request for Information, internal routing of messages, message creation (automatically or interactively), data normalization, retrospective search, and error handling. It is a generic, table driven processor that accepts formatted and unformatted USMTF-like messages from a communications front end, validates

message format and field content, then performs additional processing as directed by the user.

- Office Automation Services. Functional capabilities include word processing, email, presentation graphics, spreadsheet, drawing, illustration, and other office tools identified as necessary within the DII COE.
- On-Line Support Services. Provide [GCCS COE] users with the necessary assistance in all aspects of system operation. Four basic support services are required to achieve comprehensive coverage of system operation: On-Line Help, On-Line Job Planning, On-Line Reference, and Computer Based Instruction (CBI).

Software Development Services

- Developer's Kit. Tools, documentation, and other information items to develop GCCS compliant software. For GCCS V3.0, the Developer's Kit will focus on APIs, segmentation tools, run-time integration, User-System Interface Style Guide, and compliance metrics. It will gradually increase its focus to include computer-assisted software engineering (CASE) tools, static code checking, coding standards, security rules, reengineering tools, and other items relating to new software or to software reengineering to comply with the GCCS objective (client-broker-server) architecture.

Technology

The current DII COE is populated with legacy components adopted as "best of breed" from the military services. The principal platforms are UNIX systems from Sun and Hewlett-Packard. The graphical desktop product on UNIX platforms is TED from Triteal, an implementation of the Common Desktop Environment (CDE). The DII COE also supports Microsoft Windows NT as a desktop. Informix, Oracle, and Sybase DBMSs are in the COE. DCE is being deployed this summer for developer use and administrator training. It will become operationally available in Version 3 toward the end of 1996. The DCE facilities will include threads, remote procedure call (RPC), distributed time service, and directory service (global and cell). Table 2-1 summarizes the makeup of DII COE 2.0. A detailed list is in Appendix A.

IBM, SGI, and Digital are being added to the list of UNIX platforms for Version 3.0. CORBA is viewed as an inevitable future component, perhaps beginning to appear in Version 4.0 in 1997. The Air Force and the Navy have registered a few early CORBA requirements with DISA. The DII Distributed Computing Working Group is seeking stronger, more detailed expressions of need from services and agencies. The beginnings of a CORBA requirement are in the current SRS for Distributed Computing services (see Appendix C).

An appendix to the I&RTS is a draft version of compliance specifications for NT-based platforms running Windows. The CORBA specification in the draft SRS requires interoperation with DCOM environments.

Table 2-1. DII COE 3.0 Components

DII COE Service	Products
Operating System	Unix (HP-UX, Sun Solaris); Windows NT
Desktop	CDE TED; Windows NT
Windowing	X Window & Motif; Windows NT
Distributed Computing and Object Management	DCE Client for Unix and Windows NT (threads, RPC, time, directory) DCE Server (security, directory) Distributed File Server [DCE] Cell Manager News Make Group
Printing	[COE] Print Services
Security	Native to OS; also Console, Deadman, Password, X Display Manager (XDM), Security Manager
System Management	MENUEXEC
Data Management	Oracle, Sybase, Informix, JCALS GDMS
Mapping, Charting, Geodesy, & Imagery	JMTK
Message Processing	Internet Relay Chatter; Mail Services; Tool Command Language; Common Message Processor
Office Automation	WABI; Netscape Web Browser; Netscape News Server; NETSITE Server; MS Office
Management	FTPTool, GZIP, PERL, SPI, STREAMS, NetMatrix, Empire System Management Agent for Solaris and HP-UX, Courtney, Crack, SATAN, TCP Wrappers, Tripwire, Tivoli, NewsPrint

Source: DII COE Baseline Specification Version 3.0, October 31, 1996 (Draft)

Requirements

Technical requirements are being developed in the form of an SRS for each of the service areas. The current SRSs are drafts in progress and are oriented toward DII COE Version 4.0.

Data Architecture

The DII COE requires that data of a given category be in a category-specific format. For instance, the DII COE Track Server that provides track category data will convert the track data formats from all external sources into a category-compliant format.

A Common Data Server (CDS) is being defined. In addition, the Shared Data Environment (SHADE) is being developed. SHADE is a strategy and mechanism for data sharing that represents an extension of the principles of the DII COE. SHADE is an infrastructure that improves systems interoperability and data sharing through better management of metadata, data access mechanisms, and physical data. SHADE will provide the necessary data architectures, approaches, reusable components, and guidelines and standards for developers to field systems that will meet the user's requirements for timely, accurate, and reliable data. SHADE benefits interoperability by providing reference data segments, shared databases, configuration management of shared databases, reduction of external interfaces, and data access infrastructure. Currently, there is a lack of consistent terminology between CDS and SHADE. CDS and SHADE will come together as Shared Data Servers (SDS) in the future.

Software Integration and DII COE Compliance

The I&RTS document describes four categories of compliance—Runtime Environment, Style Guide, Architectural Compatibility, and Software Quality. The categories are described in Table 2-2.

Table 2-2. COE Compliance Categories

Compliance Category	Definition
Category 1: Runtime Environment	The degree to which a software component operates as intended within the COE, uses COE services where appropriate, and does not interfere with other components
Category 2: Style Guide	The degree to which a software component is consistent with COE look and feel guidelines and practice for the user interface
Category 3: Architectural Compatibility	The degree to which a software component fits within the COE architecture, to include a client/server structure, a DCE infrastructure, and a CDE desktop
Category 4: Software Quality	The degree to which a software component has favorable size, complexity, functional integrity, and other common metrics of software quality

Category 1, Runtime Environment, is then expanded into eight levels of compatibility. Compliance checklists are given for each level. The levels are listed in Table 2-3. The goal is to package all software as COE segments for configuration management and deployment. Another goal is to avoid duplicating any service that is already part of the COE. It is understood that early implementations of the DII COE will fall short of the ideal (Level 8 compliance). Level 5 is the minimum degree acceptable to DISA for software being offered for prototype activities in a DII COE environment.

2.1.4 Phased Implementation

The current DII COE is Version 2. Version 3 will be deployed in October 1996. Version 4 is planned for early 1998. One view of this schedule is in Figure 2-3.

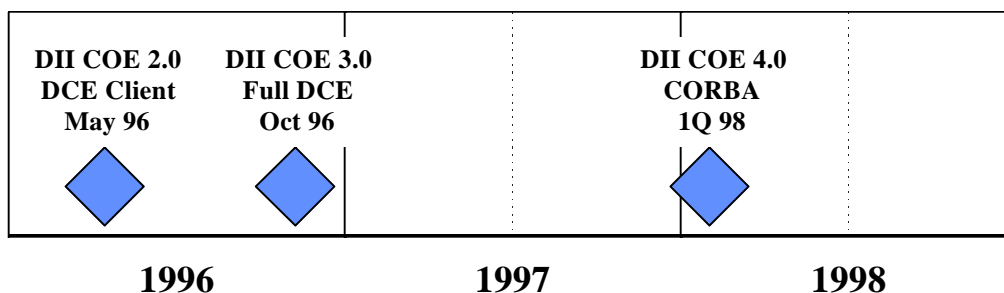


Figure 2-3. DII COE Schedule of Releases

2.2 Joint Mapping Toolkit (JMTK)

The MCG&I component of the DII COE is the service area most closely related to the USIGS. The Joint Mapping Toolkit (JMTK) is the initial implementation of those services in the DII COE.

2.2.1 JMTK Program

GCCS/JMTK is a program sponsored by the Defense Mapping Agency (DMA) (now part of NIMA) to integrate DoD software into a toolkit which will meet the MCG&I requirements of the DISA GCCS and DII COE. The GCCS/JMTK will provide standard mapping charting, geodesy, and imagery data and exploitation capabilities as a functional area for the GCCS COE. The GCCS/JMTK will be implemented through an evolutionary migration process. Version 3.0 of the GCCS/JMTK consists of an integration of the Navy's CHART product for visual capabilities, the Air Force's Common Mapping Tool Kit (CMTK) for spatial database

Table 2-3. COE Levels of Runtime Compliance

Runtime Compliance Level	Description
Level 1: Standards Compliance	A software component uses the same standards as another software component, but data sharing is undisciplined and COE facilities are not used; the software components may execute simultaneously without conflict.
Level 2: Network Compliance	Two software components coexist on the same LAN but are on different CPUs; limited data sharing is possible.
Level 3: Workstation Compliance	Two software components reside on the same LAN, share data, and coexist on the same workstation, but do not use COE services and may not be interoperable.
Level 4: Bootstrap Compliance	Software is structured as COE segments and uses the COE bootstrap segment; COE services are not used.
Level 5: Minimal COE Compliance	A software segment uses the same kernel COE as co-resident segments; boot, background, and local processes are specified through segment descriptor files; segment is registered and available through on-line library; applications appear integrated to users, but there may be duplication of function and lack of interoperability; segment may be installed and removed with COE tools.
Level 6: Intermediate COE Compliance	A software segment uses existing account groups and reuses COE component segments; minor documented differences may exist between the Style Guide and the segment's GUI.
Level 7: Interoperable Compliance	A software segment reuses COE component segments to ensure interoperability, to include communications interfaces, message parsers, database tables, track data elements, and logistics services; published APIs are used primarily, with documented use of few or no private APIs; no COE component segment function is duplicated.
Level 8: Full COE Compliance	A software component is completely integrated: makes maximum possible use of COE services and is available via the Executive Manager, is fully compliant with the Style Guide and uses only published public APIs; does not duplicate any function implemented elsewhere in the COE or in another application component.

management services, and the Army's Terrain Evaluation Module (TEM) for analysis functions. In subsequent releases, the GCCS/JMTK will migrate from a constrained approach driven by its initial reliance on the Service components' software contributions to an independent architecture that is objective and in compliance with the DII COE.

2.2.2 JMTK Description

GCCS/JMTK [15] is a mapping and analysis system being implemented to provide a variety of services and a communications backbone. It consists of computer software running on UNIX workstations. GCCS/JMTK components are combined to yield specific configurations designed to support a variety of users at various locations. The JMTK is an open system capable of running on any GCCS COE approved platform. The current, approved Commercial Off-the-Shelf (COTS) hardware platforms for GCCS are Hewlett Packard (HP) 9000/700 series workstations and Sun SPARC 10/20/1000/2000 series workstations running under UNIX, and client hardware platforms running under Windows NT (with servers under NT soon to be included). Presently the GCCS COE version 2.0 software platforms are Sun Solaris version 2.5 and HP UX version 9.0.7 with OS compatible versions of X-Windows and Motif. Ultimately, the GCCS/JMTK is to be hardware-independent and operate on a range of open system platforms running under standards-based operating systems designated by GCCS.

The GCCS/JMTK provides common geospatial processing and data to all mission applications and users within the GCCS COE. GCCS/JMTK services consist of a set of geospatial processing components, some of which interface with other GCCS COE elements. GCCS/JMTK provides visualization, analysis, and spatial database management software capabilities for standard geographic information types. GCCS/JMTK functional services are divided into the following seven functional areas, or domains:

- Domain 1 - Spatial Database Management (of DMA products as well as files generated by GCCS/JMTK),
- Domain 2 - Visual (display of maps and areas of interest),
- Domain 3 - Analysis (e.g., terrain analysis, line of sight),
- Domain 4 - Utilities (e.g., housekeeping, error messages),
- Domain 5 - Local Image Manipulation (e.g., satellite, photograph),
- Domain 6 - Overlay Manager; formerly referred to as Geospatial Data Services (e.g., building of overlays, storing of preferences), and
- Domain 7 - Security, Access and Data Releaseability.

2.2.3 JMTK Schedule

The JMTK development schedule includes a progression of point releases of what is substantially the present implementation, followed in the longer term by the JMTK Objective Architecture. The JMTK Objective Architecture will be based on distributed object technology.

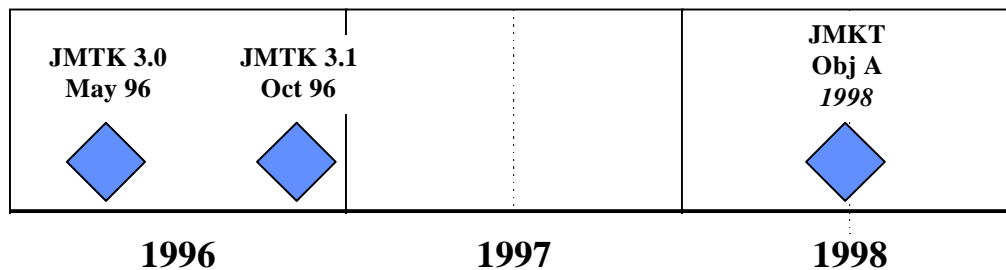


Figure 2-4. JMTK Schedule

2.3 Military Intelligence Database (MIDB) and Image Products Archive (IPA) for GCCS (MIG)

As an implementation of MCG&I services for the DII COE, initial versions of the JMTK may be viewed as having more to do with maps, charts, and geodesy and less with imagery, with a goal of achieving a more appropriate balance over time. Recently, an accelerated effort to provide improved imagery support in the near term has been undertaken by the Joint Staff. Called MIG (Military Intelligence Database (MIDB) and Image Products Archive (IPA) for GCCS) [16], it addresses access to and integration of intelligence and imagery data into the GCCS. MIG includes tools for merging imagery, maps, and the MIDB into the consistent operational picture (COP). MIG entails creation of Image Transformation Services (ITS) for image cataloging and storage, image transformations, links to the MIDB, merging imagery with maps, and imagery and track overlays.

MIG development is scheduled in three phases, as shown in Figure 2-5. Video handling is a major area of development in Phases 2 and 3.

2.4 USIGS Architecture and CIIF Reference Model

As background for the CIIF Reference Model, consider the draft Intelligence Community Reference Model shown in Figure 2-6. The Intelligence Community Reference Model represents distributed object technology in the context of the POSIX and TAFIM style of technical reference model. It retains the three types of entity of the POSIX model—

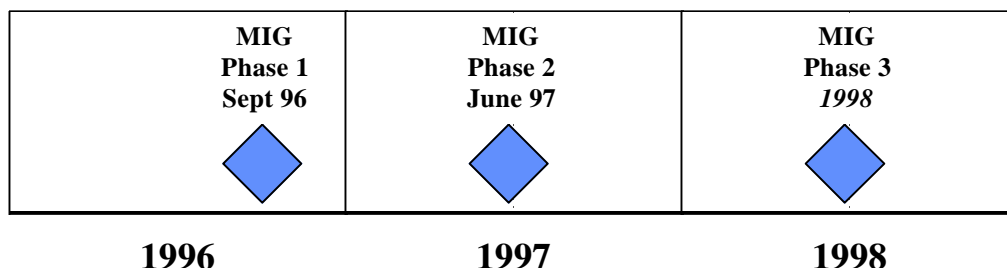


Figure 2-5. MIG Schedule

application software, platform, and external. It retains the four types of POSIX service interface—human-computer, system, information, and communications. But it adds an application programming interface for object services. Also, it makes object services an explicit component of the platform entity. Relative to the TAFIM (discussed in the next section), it restructures the application software entity into three classes of application—mission specific, support, and common facilities (“common facilities” is a term taken from the OMA).

2.4.1 Objective of CIIF

As further background to the CIIF Reference Model, consider the USIS Architecture³, of which the CIIF is a part. A view of the USIS Architecture as a set of interoperating digital elements is shown in Figure 2-7. The services accessed through CIIF interfaces are designed to enable USIS digital elements to interoperate.

The CIIF Reference Model, shown in Figure 2-8, specifies a framework for developing an open application program interface (API) between architectural elements of the USIS. It focuses on services provided inside the boundaries of the USIS, and even more specifically on those interfaces that require standardization within the USIGS. The CIIF Reference Model identifies interfaces that address related API functions, and groups them into interface architecture building blocks called “facilities.”

At a more detailed level, Common Facilities and Imagery Interfaces are defined as those interfaces and uniform sequencing semantics that are shared across applications in such a way

³ Although the USIS architecture is now considered to be part of the NIMA USIGS architecture, in which geospatial processing is added to the imagery processing addressed by USIS, this development is too new to be useful for the present discussion.

as to make object-oriented distributed computing applications much easier to create. Common Facilities and Imagery Interfaces comprise both generic facilities and domain-specific

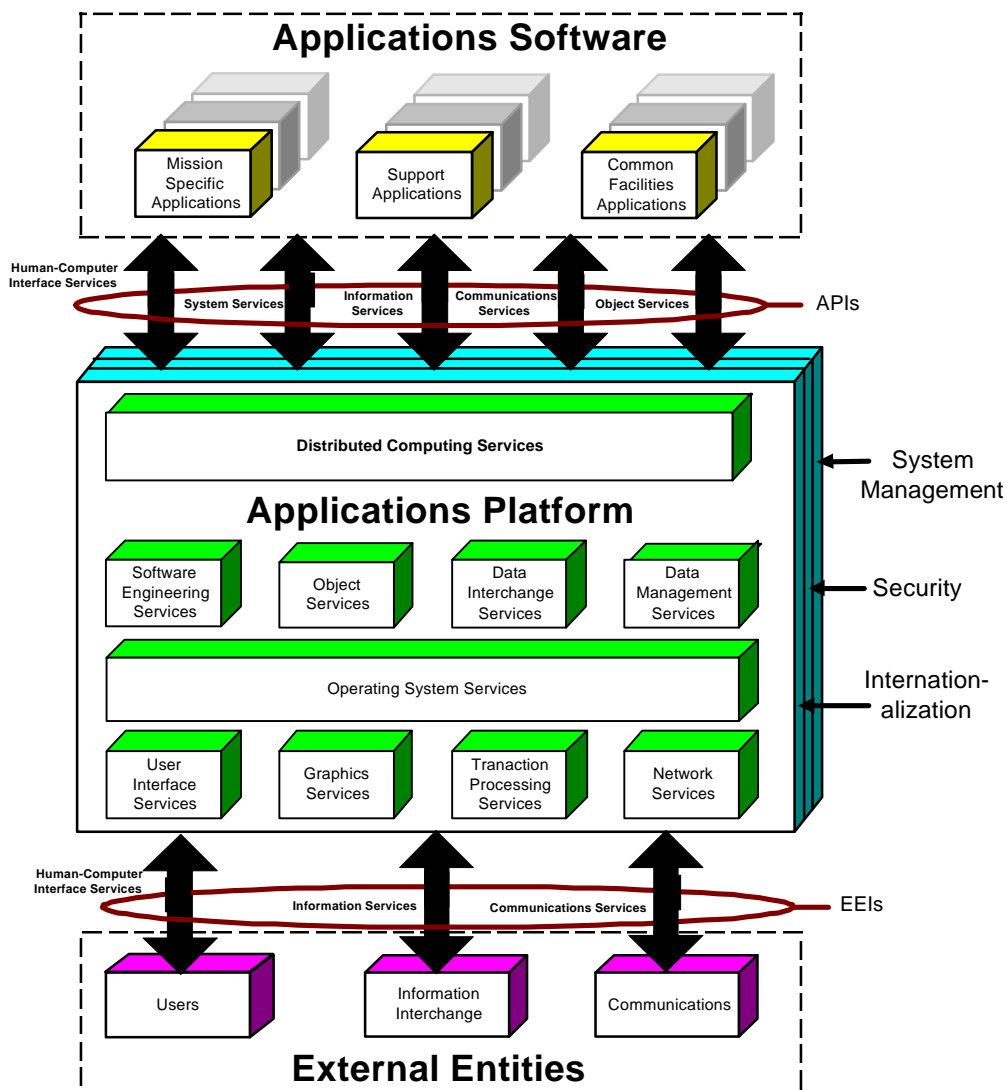


Figure 2-6. Intelligence Community Reference Model (DRAFT)

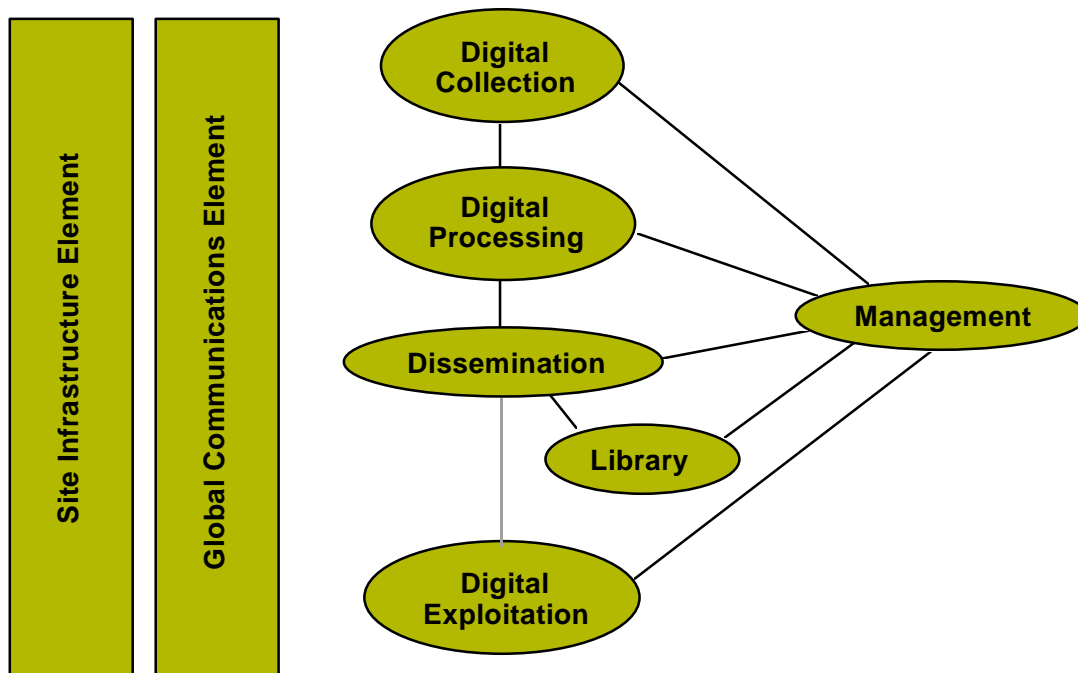


Figure 2-7. USIS Architecture—Digital Elements

specifications. Examples of the kinds of inter-application services provided by Common Facilities and Imagery Interfaces include object cataloging and browsing, help facilities, object rendering, printing and spooling, and objects which implement generic business rules for the imagery industry.

2.4.2 Services and APIs

The CIIF Reference Model [17] is designed to depict an object-oriented infrastructure, in which CIIF functions are accessed through an object request broker. It classifies the components, interfaces, and protocols that comprise an object system, following the Object Management Architecture (OMA) of the OMG. The CIIF reference model has six key components:

- Distributed Computing Infrastructure — Enables software objects to make and receive requests and responses within a distributed environment
- Object Services — A collection of fundamental services (interfaces and objects) that provide basic functions for using and implementing other software objects

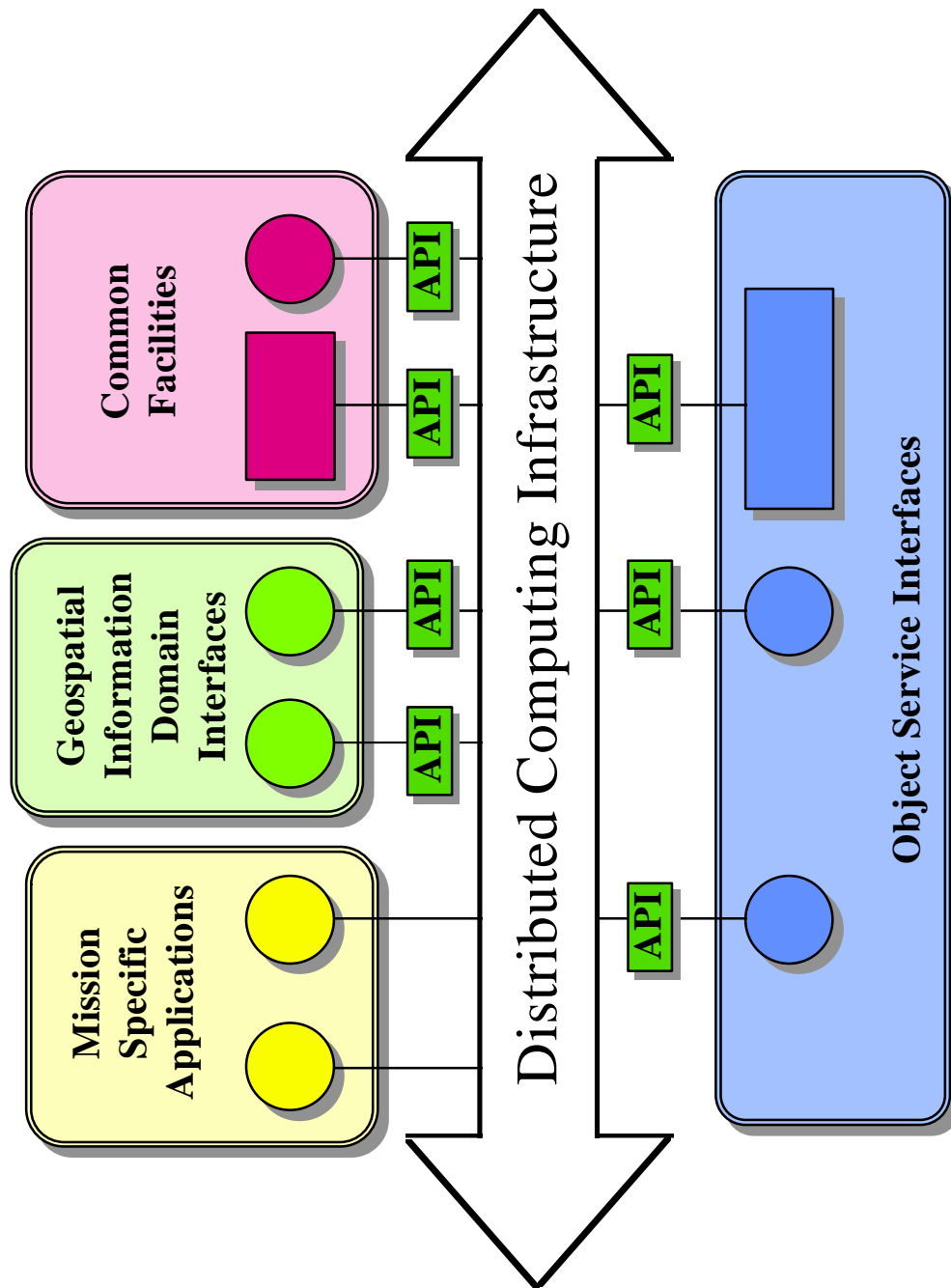


Figure 2-8. CIIF Reference Model

- Common Facilities — A collection of higher-level services that are broadly usable by many applications
- Imagery Interfaces — Standard interfaces that promote object-based interoperability within the imagery community or application domains
- USIGS Applications — Software objects specific to the USIGS, including particular commercial products or end-user systems
- Interface Definition Language (IDL) — A formal language, defined by OMG and being standardized by the International Organization for Standards (ISO) [18], that is used to define the interfaces between interoperable software objects.

Analysis of the USIS Technical Architecture Requirements [19], coupled with an effort to apply the CIIF's distributed computing architecture principles, has led to definition of the following facilities (see *CIIF Reference Model* for further description):

- Catalog Access Facility — Supplies a set of common software interfaces to support both local and global imagery product discovery, product attribute (metadata) retrieval, product browsing, and product cataloging and indexing
- Image Access Facility — Defines a set of interfaces for retrieving select imagery products, including video and video-derived products, from an imagery library, and for updating the contents of an imagery library (by storing, deleting, or modifying imagery products)
- Imagery Dissemination Facility — Defines the interfaces required to receive, prepare (i.e., reformat, compress, decompress, etc.), prioritize, and transmit imagery products; also defines standard interfaces to support product distribution management
- Profile and Notification Facility — Supplies a set of standard interfaces to support the registration and maintenance of standing interest profiles for imagery consumers; also provides interfaces to support the screening of products against these profiles, and to route products or product availability notifications, as appropriate

Imagery exploitation is fundamental to the USIS. It leads to the generation of intelligence reports and other products which ultimately reach policy makers and other consumers of intelligence. Additional exploitation facilities are likely to be added to this reference model. The following list represents the current proposed categories of facilities for imagery exploitation services:

- Image Annotation Facility—Provides standard interfaces to software tools that enable symbols, graphics, text, and other media types to be overlaid upon imagery to highlight significant content.

- Image Manipulation Facility — Provides interfaces to standard algorithms for manipulating imagery (resizing, changing color and contrast values, applying various filters, manipulating image resolution, etc.) and for conducting mathematical analyses of image characteristics (computing image histograms, convolutions, etc.)
- Image Mensuration Facility — Provides standard interfaces to software tools that are designed to measure the spatial characteristics of objects appearing within images
- Image Registration Facility — Provides standard interfaces for automatically aligning, co-registering, or otherwise determining image-to-image spatial correlations on the basis of image content
- Geolocation Facility — Defines standard interfaces to software tools that support the derivation of precise geographic coordinates on images and maps
- Automatic Target Recognition — Provides standard interfaces to software tools that are designed to automatically detect, categorize, count, and determine relationships between objects appearing within images
- Image Synthesis — Provides a common software interface for creating or transforming images using computer-based spatial models, perspective transformations, and manipulations of image characteristics to improve visibility, sharpen resolution, and/or reduce the effects of cloud cover or haze
- Image Understanding — Enables automated image change detection, registered image differencing, significance-of-difference analysis and display, and area-based and model-based differencing

The following Common Facilities, which are likely to be developed by other organizations such as OMG, fulfill key requirements of the USIGS Technical Architecture. As the IDL specifications for each of these facilities are completed and published. They will be thoroughly evaluated, and those that are found to meet the requirements for the USIGS will be adopted as components of the CIIF.

- Automation and Scripting Facility — Defines conventions and interfaces that allow access to the key functionality of an object from another object. The design goal of this facility is to support user visible objects which are larger grained than the typical ORB object. The typical object acted upon by the Automation and Scripting Facility would be a document, a paragraph, a spreadsheet cell, and so forth. The emphasis of the facility is for objects to expose enough of their capabilities so they may be driven by scripts and macros.
- Common Management Facility — Provides a set of utility interfaces for system administration functions. These abstract basic functions such as control, monitoring, security management, configuration, and policies that are needed to perform systems management operations, such as adding new users, setting permissions, installing software, and so forth.

- Compound Presentation and Interchange Facility — Enables the creation of cooperative component software that supports compound documents, that can be customized, that can be used collaboratively, and that is available across multiple platforms. Also provides for the storage and interchange of data objects, and roughly maps to the persistent storage subsystem of a compound document architecture.
- Data Interchange Facility — Allows for the exchange of information across networks of heterogeneous computer systems by providing a common information model and a common way of encoding information within that model. Encoding must support not only character data, but other sorts of data as well, including imagery, graphics, multimedia documents, and electronic mail. Enables objects to interoperate through exchange of data, and can be used for many forms and kinds of data transfer, such as: bulk transfer; interchange of formatted data such as TIFF, GIF, EPS, NITF, etc.; structured data transfer such as ISO IDL specified data types; interchange of domain-specific object representations; and the data interchange between objects and encapsulated software (legacy applications).
- Imagery Compression Facility — Defines a set of interfaces to generalized services for imagery compression and decompression, and for conversion between internal representations and standardized representations of such data
- Information Storage and Retrieval Facility — Comprises the higher level storage and retrieval specifications for distributed applications. These specifications will be applicable to a wide range of information services, including data base access and information highways.
- Internationalization and Time Operations Facility — Enables developers to use an information system or application in their own language using their own cultural conventions. In addition, this technology will allow the developer to use a culture's numeric and currency conventions, and keep track of time zones.
- Meta-Object Facility — Defines the interfaces and sequencing semantics needed to create, store and manipulate object schemas that define the structure, meaning, and behavior of other objects within the OMG Object Management Architecture. These objects may be application objects, common business objects, objects representing analysis and design models of applications, or objects providing the functionality of Common Facilities and Common Services. The Meta-Object Facility can be used in an information system (such as a repository) that enables an enterprise to specify and manage a wide variety of information assets with a common, integrated set of services. The use of a common meta-object facility for specifying the schemas of the information

assets will play a key role in helping to achieve data and process integration by enabling tools and processes to share information and coordinate activities.

- **Mobile Agents Facility** — Supports the need to create massively distributed information systems over Wide Area Networks. Agent technology efforts range from building these massively distributed systems to mobile information systems, intelligent workflow systems, and agile corporation information structures.
- **Printing Facility** — One component of a coordinated set of facilities and standards needed to satisfy the printing requirements of the modern distributed office. Together, the capabilities provided must enable users to create and produce high-quality documents in a consistent and unambiguous manner within a distributed object environment. The Printing Facility should be able to meet a range of printing requirements from simple one document, one copy printing, all the way up to high volume production printing, which might involve several documents, several copies, several printers.
- **Rendering Management Facility** — Provides facilities to present information for output on devices such as screens, printers, plotters and sound and speech output devices. It also handles user input from a variety of hardware devices such as a mouse, keyboard, scanner, speech recognition device, digital camera, and security devices. Rendering management includes support for window management, class libraries for user interface objects, user interface dialog objects, and abstractions of the many different input/output devices.
- **Security Administration Facility** — Provides standard interfaces, as well as the necessary control mechanisms, to facilitate required security protections, including provisions for:
 - User registration, password maintenance, permissions maintenance
 - Access control, authentication, and audit trail maintenance
 - Resource registration
 - Security classification downgrading
 - Encryption key management
 - Discretionary and mandatory access control.

Workflow Facility — Provides management and coordination of objects that are part of a work process for example, purchase orders. The facility will provide support for production-based workflow, which is structure, pre-defined processes that are governed by policies and procedures, as well as ad-hoc, or coordination-based workflows, which are evolving workflows defined by one or more people to support the coordination of knowledge workers.

2.4.3 Interfaces that Comprise a Facility

A facility may have several distinct interfaces (i.e., it may define multiple semantically-related interface types). A taxonomy of these interfaces is presented here, because it is important in characterizing facilities to clearly distinguish what interfaces are involved in providing a facility, how they relate to each other, how one gets access to them, and who is expected to use them. The interfaces to a facility can be characterized by:

- **Audience** – The types of the anticipated consumers (callers) of an interface. An interface may be intended for use by the ultimate user of the facility or it may be intended for use by a system management function within the system. In more complex facilities, objects whose function and implementation lie completely outside of the facility may need to collaborate to fulfill the original facility's functions. In this case, interfaces may be defined that are used to construct the facility from a series of disparate objects. The audience for such interfaces is neither the user of the facility nor a system manager, but rather the other objects that participate in creating the facility.
- **Bearer** – The object type that presents an interface. An object may be fundamentally characterized by the fact that it has a given interface, or an object may have an interface that is ancillary to its primary purpose (in order to provide certain other capabilities).

The term *audience* characterizes who (or what) uses the different interfaces that comprise a facility. Such interfaces can be categorized as belonging to one of three classes according to their intended audience:

- **Functional Interfaces** – Interfaces that define the operations invoked by the primary consumers or users of the facility. These interfaces present the functionality (the useful operations) of the facility. A given facility may have several functional interfaces to provide different aspects of its overall collection of services.
- **Management Interfaces** – Interfaces used to communicate with system management services and facilities. These interfaces handle operational control of a service (e.g., setting threshold levels), as well as its installation and deployment (e.g., starting and stopping a service).
- **Construction Interfaces** – Interfaces that define the operations used to communicate between the core of a facility and related objects that participate in providing the service. These interfaces are typically defined by the facility, and then inherited and implemented by participants in the facility. In other words, these interfaces are invoked by the facility provider itself. Objects that participate in a facility must support these interfaces. A given facility may have several construction interfaces to connect various parts of its implementation.

The term *bearer* characterizes the objects which present a particular interface. The bearer of an interface can be further categorized according to whether that interface defines the core function of the object (i.e., a specific object bears the interface) or whether that interface

defines additional capabilities for an object whose core purpose is something else (i.e., some generic object bears the interface); that is,

- Specific objects can bear an interface. By a specific object it is meant an object whose purpose for existence is to constitute that part of the facility whose interface it carries. The notion is that a limited number of implementations (and potentially a limited number of instances) of these objects exist in a system, usually as “servers.”
- Alternatively, generic objects can bear an interface. In this context, a *generic object* is an object whose primary reason for existence is unrelated to the facility whose interface it carries. The notion is that the facility is provided by having any of several other object types inherit and implement that facility’s interface.

2.4.4 Phasing of CIIF and IDL Development Activities

The Common Imagery Interoperability Working Group (CIIWG) defines priorities and recommends schedules for CIIF development activities. Figure 2-9 shows the schedule for developing elements of the CIIF. The CIIF Reference Model (RM) is being revised and will appear as Version 2 in December 1996. After that, it is envisioned that further definition of the CIIF will be done within the framework of the NIMA Technical Architecture. The

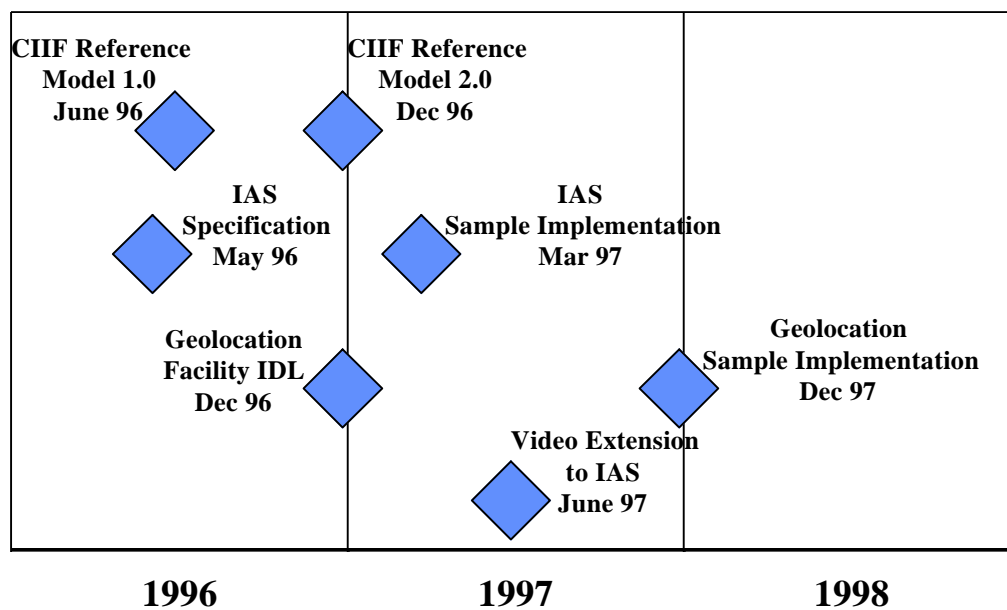


Figure 2-9. Schedule of CIIF Development Activities

Image Access Services CIIF (Image Access, Catalog Access, Profile Notification, Imagery Dissemination) have been defined in ISO IDL; a sample implementation will be developed to validate and refine the IDL specification. The Geolocation CIIF are the next scheduled for definition in ISO IDL, then sample implementations. A further line of development involves the addition of video services to the Image Access Services CIIF; the MIG effort described earlier is part of the video development activity. Development schedules for other CIIF functions have not been proposed, but are expected to span the remainder of the decade.

2.5 Standards Profiles and Technical Architectures

As shown in Figure 1-1, there are several architectural activities that affect the DII COE. Five of them—TAFIM, JTA, DoDIIS, USIGS Standards and Guidelines, and Intelink—are summarized here.

2.5.1 TAFIM Reference Model and Standards Profile

The purpose of the Department of Defense Technical Architecture Framework for Information Management [6,7] is to provide guidance for the evolution of the DoD information processing infrastructure. The TAFIM does not define a specific system architecture. Rather, it identifies services, standards, design concepts, components, and configurations that can be used to guide the development of technical architectures that meet specific mission requirements. The TAFIM is an Enterprise Level guide for developing technical architectures. Integrating functional and technical requirements of DoD information systems can be portrayed using the DoD Information Management integration model shown in Figure 2-10. It represents a perspective for defining boundaries for potential integration pay-off from a DoD-wide viewpoint. Further, it can assist integrators in defining what is to be integrated in order to correctly proceed with the task. Functional and technical integration requirements must be addressed both at the vertical boundaries within a level and the horizontal boundaries between the levels of the model.

The purpose of the TAFIM Technical Reference Model (TRM), Volume 2, is to provide a common conceptual framework, and define a common vocabulary so that the diverse components within the DoD can better coordinate acquisition, development, and support of DoD information systems. The TAFIM TRM, shown in Figure 2-11, also provides a high-level representation of the information system domain showing major service areas. DoD organizations are required to apply the model to increase commonality and interoperability across the DoD. The model is not a specific system architecture. Rather, it establishes a common vocabulary and defines a set of services and interfaces common to DoD information systems. The reference model and standards profile define the target technical environment for the acquisition, development, and support of DoD information systems. The objectives to be achieved through application of the TAFIM TRM are to improve user productivity,

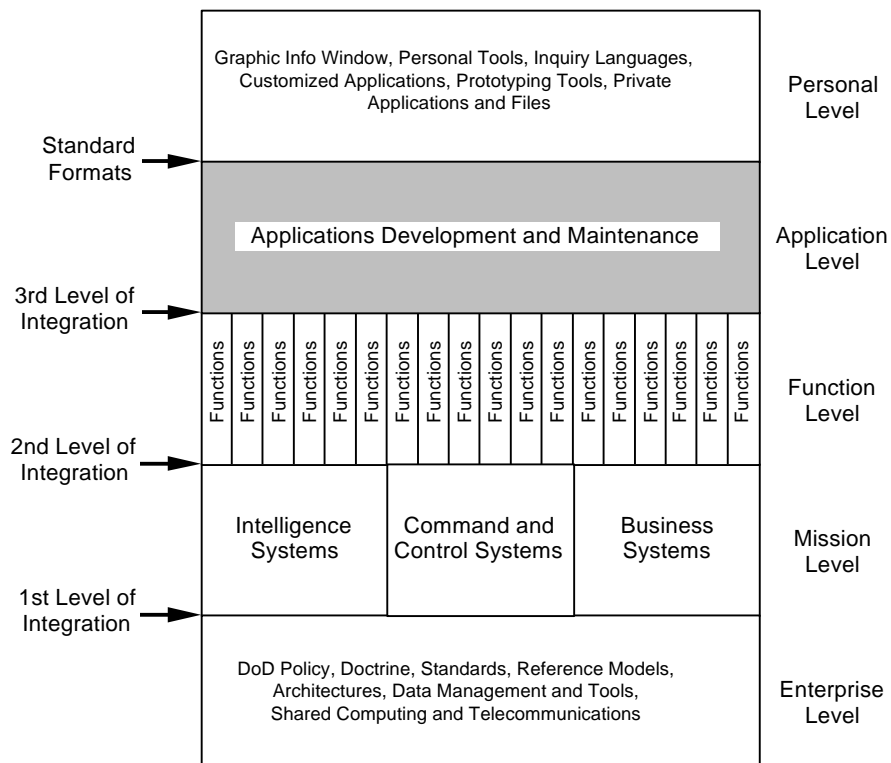


Figure 2-10. DoD Information Management Integration Model (TAFIM V2.0 Vol. 1)

development efficiency, portability, scalability, interoperability, security, manageability, vendor independence, and life-cycle costs.

The purpose of the TAFIM Adopted Information Technology Standards (AITS), Volume 7, is to guide DoD Enterprise acquisitions and the migration of legacy systems by providing a definitive set of information technology (IT) standards to be used in DoD. These standards provide consistency across the Enterprise, Mission, Function, and Application levels of the DoD Integration Model.

2.5.2 Joint Technical Architecture (JTA)

The Joint Technical Architecture (JTA) [9] identifies a common set of mandatory information technology standards and guidelines to be used in all new and upgraded command, control, and intelligence systems and the communications and computers that support them (C4I) in support of the Warfighter battlespace. These guidelines consist primarily of a common set of standards/protocols to be used for sending and receiving information, for understanding information, and for processing that information. The JTA also includes a

common human-computer interface and a set of information system security standards for protecting the information.

The standards and specifications identified in the JTA are entirely consistent with the general guidance provided in the TAFIM. The JTA will be used by anyone involved in the management, development, or acquisition of new or improved C4I systems within DoD. While the strategy for implementation is being formulated and discussed now, the guiding principle generally agreed to is that the responsibility for specific implementation details, enforcement decisions and mechanisms will be determined by each of the Services and Agencies Acquisition Executives (SAEs). System developers will use the JTA to ensure that new and upgraded C4I systems meet interoperability requirements. System integrators will use the JTA to facilitate the integration of existing and new systems. Developers of operational requirements will take cognizance of the JTA in developing requirements and functional descriptions. The DoD science and technology community will use the JTA during

the design phase to ensure that their concepts will readily integrate into existing systems and increase the likelihood of interoperability.

2.5.3 DoD Intelligence Information System (DoDIIS)

The Department of Defense Intelligence Information System (DoDIIS) Technical Reference Model [4] shown in Figure 2-12 establishes an architectural framework intended to foster transition to a standards-based open system architecture within the DoDIIS Community. It is an adaptation of the TAFIM to DoDIIS. DoDIIS platform services are designed to be accessed at interfaces that make the implementation-specific characteristics of the platform transparent to application software. A profile of APIs provides guidance to DoDIIS community members responsible for the procurement of hardware and software for the upgrade of existing intelligence capabilities and the implementation of new capabilities at DoDIIS sites. The current DoDIIS infrastructure architecture is known as Client-Server Environment System Support (CSE SS).

DoDIIS systems implemented in accordance with the DoDIIS API profile will support overall DoDIIS goals to improve user productivity, provide or improve interoperability across intelligence functions and DoDIIS, improve development efficiency across the DoDIIS community, minimize or reduce life-cycle costs, and comply with security requirements. The Military Services and commands are encouraged to apply the DoDIIS API profile to lower echelons to ensure conformity across the DoDIIS community. Interoperability with tactical systems is not fully addressed in the DoDIIS API profile.

To maximize interoperability, DoDIIS has adopted the DoD Joint Technical Architecture (JTA), and is being migrated to DII COE compliance. The JTA will essentially replace the DoDIIS Profile of the DoD Technical Reference Model for Information Management as the primary technical guidance document for the DODIIS community. It is the intention of the

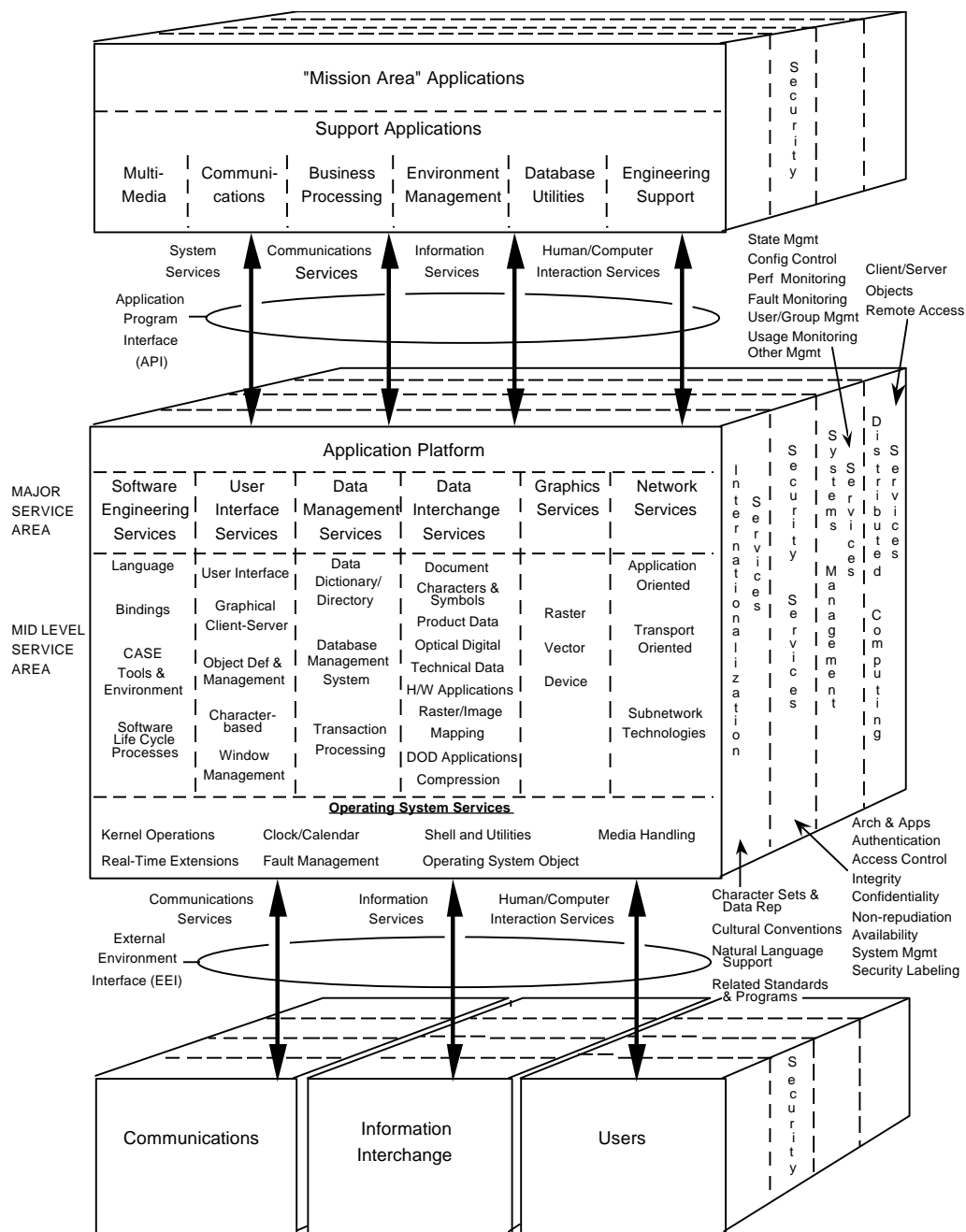


Figure 2-11. Detailed DoD Technical Reference Model (TAFIM V2.0 Vol. 2)

DoDIIS Management Board to refine and augment JTA guidance with guidance specific to DoDIIS in areas where it is deemed necessary.

The current planning envisions that DoDIIS applications will be structured into DII COE-compliant segments during 1997, with initial deployment in 1998. The target environment is DII COE Version 4.0.

2.5.4 Intelink Standards Profile

Intelink is an integrated Intelligence Information Service based on Internet technology. Intelink provides uniform methods for exchanging intelligence among providers, and between providers and users. The Intelink user community includes those who support policy analysis, foreign affairs, military operations, and law enforcement. Intelink provides a comprehensive set of tools to discover, access, and retrieve intelligence information; services to permit collaboration among analysts and/or users; and transparent utilities to improve user productivity. The basic foundations for Intelink are Internet facilities that simplify navigation and access to information; commercial information services that provide transparent access to a wide variety of services; and the movement toward a National Information Infrastructure (NII) with technology developments in communications, multimedia systems, and information access.

The operational concept for Intelink is a hybrid of the Internet and commercial services such as America On-Line and CompuServe. Like the Internet, Intelink connects users who have different technology, but adheres to common interface standards. Like the commercial services, Intelink provides its services through user interfaces ranging from text-oriented displays to graphical user interfaces (GUIs). Using Intelink standard access methods, organizations develop their own customized user interfaces and rely on sites to perform well-defined system operations roles.

The goal of Intelink is to provide any authorized user (DoD or non-DoD) access to a broad range of information sources and services through the internal system of their choice. The actual location and structure of the data will be transparent to Intelink users. To achieve this goal, Intelink adopts common standards, conventions, and procedures necessary for operation. Users and providers of data and services comply with these common items at the point of interface to Intelink. Intelink does not specify either the information sources or data to be available via Intelink, nor does Intelink own or maintain intelligence files or data. The determination regarding the sensitivity of data made available by a provider is the responsibility of that provider. Each participating organization determines who in the organization will have access to Intelink services.

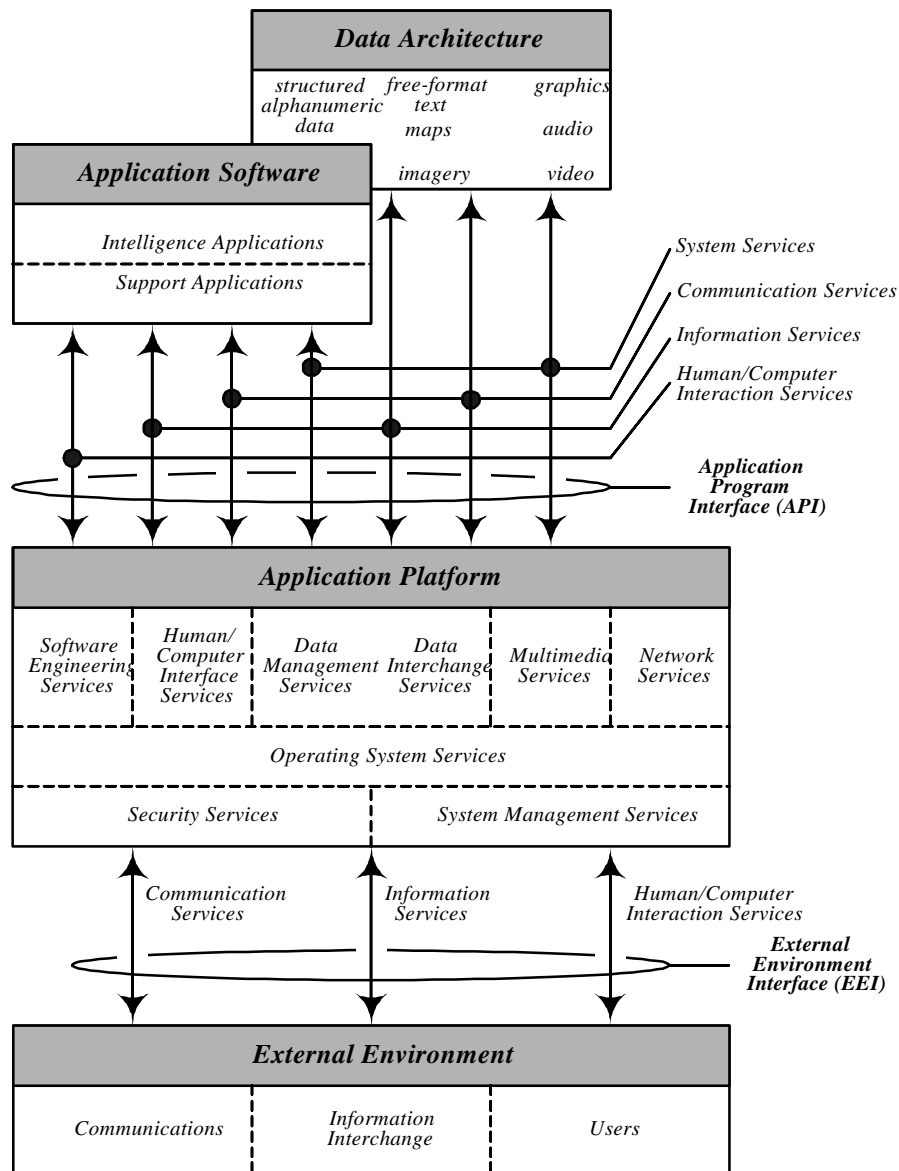


Figure 2-12. DoDIIS Technical Reference Model

2.5.5 USIGS Standards and Guidelines

The USIGS Standards & Guidelines (USIGS S&G) document [20] contains the imagery-specific standards required for any imagery-related applications of any given organization's

open systems computing environment. The USIGS S&G provides, therefore, a basis for interoperability among the systems and networks that form the USIGS. The document's

standards, conventions, and guidelines apply to the planning, design, development, test, evaluation, and operation of imagery and imagery-related systems comprising the USIGS. The USIGS S&G defines the USIGS standards profile, how it fits in the hierarchy of profiles, and how it is applied to the USIGS Architecture. The document also contains information required for its application to include: a definition of the USIGS Technical Reference Model, a forecast of advancing technology and community standards, and standards application information.

Overall, the USIGS standards profile is a dynamic part of the overlapping Information Technology communities within the US government. While the scope of the USIGS S&G is limited to imagery-specific standards in order to ensure interoperability among elements of the USIGS, other complementary standards applying to the exchanges of information and services within an element are identified in higher-level profiles (e.g., the TAFIM or Intelligence Community Standards, Conventions and Guidelines) or peer profiles such as those published by DoDIIS. The USIGS standards profile does identify imagery-specific services to be provided by the USIGS elements and the standards by which those services will be delivered.

The document specifically addresses data formats for the USIGS. In addition, the USIGS S&G addresses the important issue of service-to-service interactions. The finished IDL specifications defined in the CIIF will be referred to in the USIGS S&G.

2.5.6 Imagery Standards Management Committee (ISMC)

The ISMC is a DoD and Intelligence Community imagery standardization committee jointly chaired by DISA and the Intelligence Systems Board. The ISMC is chaired by the NIMA Systems Engineering and Program Integration Office or a designated representative. ISMC membership includes all DoD and National organizations involved in the development of systems, products, or services within the USIGS. The purpose of the ISMC is to provide the focal point for information technology standards within the Intelligence Community. The duties of the ISMC are to lead, manage, integrate, and coordinate imagery community efforts to develop and implement imagery information technology standards in information systems. The ISMC develops, establishes, implements, and promulgates new and existing imagery standards to ensure compatibility and interoperability of imagery among imagery community systems. ISMC oversees all imagery community information technology standards and interoperability activities within the scope of the NIMA. The ISMC is the configuration management authority for the USIGS S&G and the CIIF.

Section 3

Interoperability Analysis

The steps necessary to integrate the USIGS CIIF with the DII COE will be analyzed in this section. The steps proposed will account for the processes by which DISA adopts new requirements, new technology, and new services into the DII COE. The steps also will reflect the reality that the DII COE today contains support services and a set of MCG&I services that appear to overlap some of the planned CIIF services.

The DII is designated to be the common infrastructure for DoD information systems. The “integration” of an application or service with the DII COE can take either of two primary forms:

- An application operates over the infrastructure, i.e., system, network, and data services required by the application are obtained by invoking DII COE APIs
- A service is made part of the infrastructure, i.e., its functions are made available through DII COE APIs

Note that in the second case, integration of a service into the DII, aspects of the first are likely to be present as well. As DII services, the functions provided by the CIIF will be offered through DII APIs. However, the service also may require support from other DII services, and to the supporting services will look like an application. This is just another example of the now familiar idea that a server process may also be a client of other servers.

The adoption of a CIIF service into the DII COE has two aspects—API and implementation. Since a CIIF specification consists primarily of interface definitions and does not include implementation requirements, it is the API facet of DII COE adoption that is of greatest interest.

3.1 Overview of Relationship between USIGS CIIF and DII COE

The DII COE is inherently of broader scope than the USIGS. While the USIGS is a comprehensive architecture for national imagery and geospatial data collection, storage, retrieval, and exploitation, the DII COE is a Defense-wide architecture and implementation of common information system services, including operating system, data management, messaging, user interface, security, mapping, imagery, office automation, communications, directory, and interprocess facilities. The imagery functions of USIGS assume the existence of a distributed computing infrastructure and related services such as those the DII COE is intended to provide.

When comparing the CIIF to the DII COE, a further difference concerns the distinction between interfaces and implementations. Collectively, the CIIF specifications describe a high-level service architecture and the interfaces through which they can be requested. There are no

restrictions on implementation of the services beyond the need to present the specified interfaces.

On the other hand, the DII COE comprises not only a service architecture and APIs, but a reference implementation as well. The need for an early operational DII COE led to the use of legacy components that in some cases interoperate only weakly. Thus, the current DII COE constitutes only a first step toward creating a common infrastructure. In place of each military department and agency having different interface standards and implementations for the same function, the DII COE, in principle, contains only one implementation of a function. The API of the implementation then becomes the DII COE API for that service. Clearly the initial stages of this process are not likely to lead to a well structured, standards-based API architecture. This is in direct contrast to the CIIF effort.

3.2 Reference Model Analysis

To visualize how the CIIF can be integrated with the DII COE, it is helpful to consider their reference models. Figure 3-1 depicts the creation of a new reference model by merging the DII COE Reference Model (Figure 2- 2), the CIIF Reference Model (Figure 2-6), and the Intelligence Community Reference Model (Figure 2-4). The new reference model has been created to establish a more common basis for comparing them. The merged reference model shows CIIF functions in the context of a modified DII COE.

Recall that a reference model is designed to highlight *interfaces* for *services*. It does not show physical components or connections, nor does it show software modules or aspects of software implementation.

An enlarged view of the merged model is shown in Figure 3-2. A number of points should be noted. In general, it more closely corresponds to the structure of the CIIF Reference Model across the top and the POSIX model across the bottom. Other specific differences include the following:

- The COE platform now includes four classes of service:
 - Kernel
 - Infrastructure
 - Common Support Applications
 - *Common Facilities (new, from OMG Object Management Architecture (OMA) by way of the Intelligence Community Reference Model and the CIIF Reference Model)*
- The Applications layer at the top of the diagram includes the same three components as the Intelligence Community Reference Model, and they are graphically arrayed much as they are in the CIIF Reference Model (and the OMA)

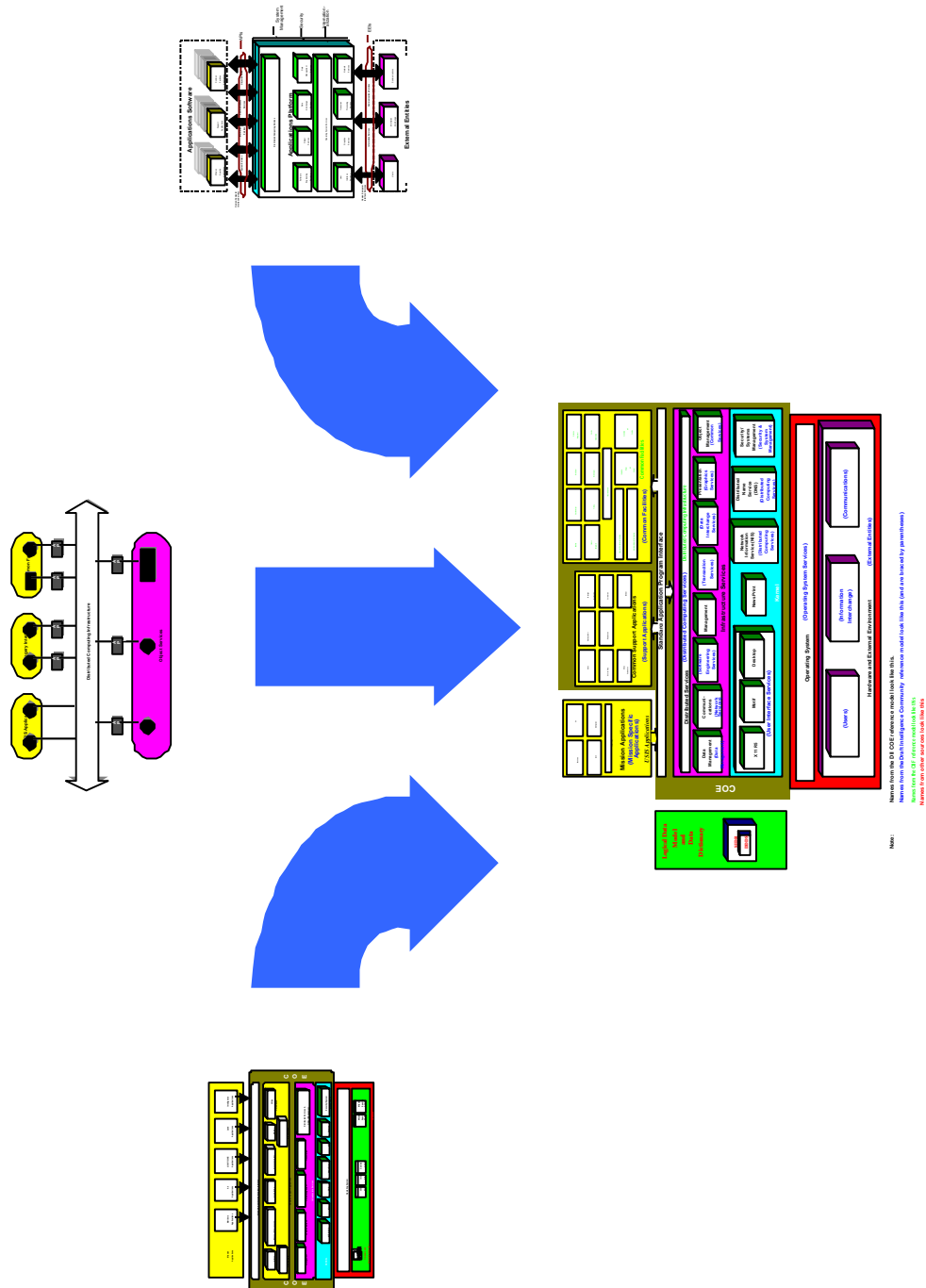
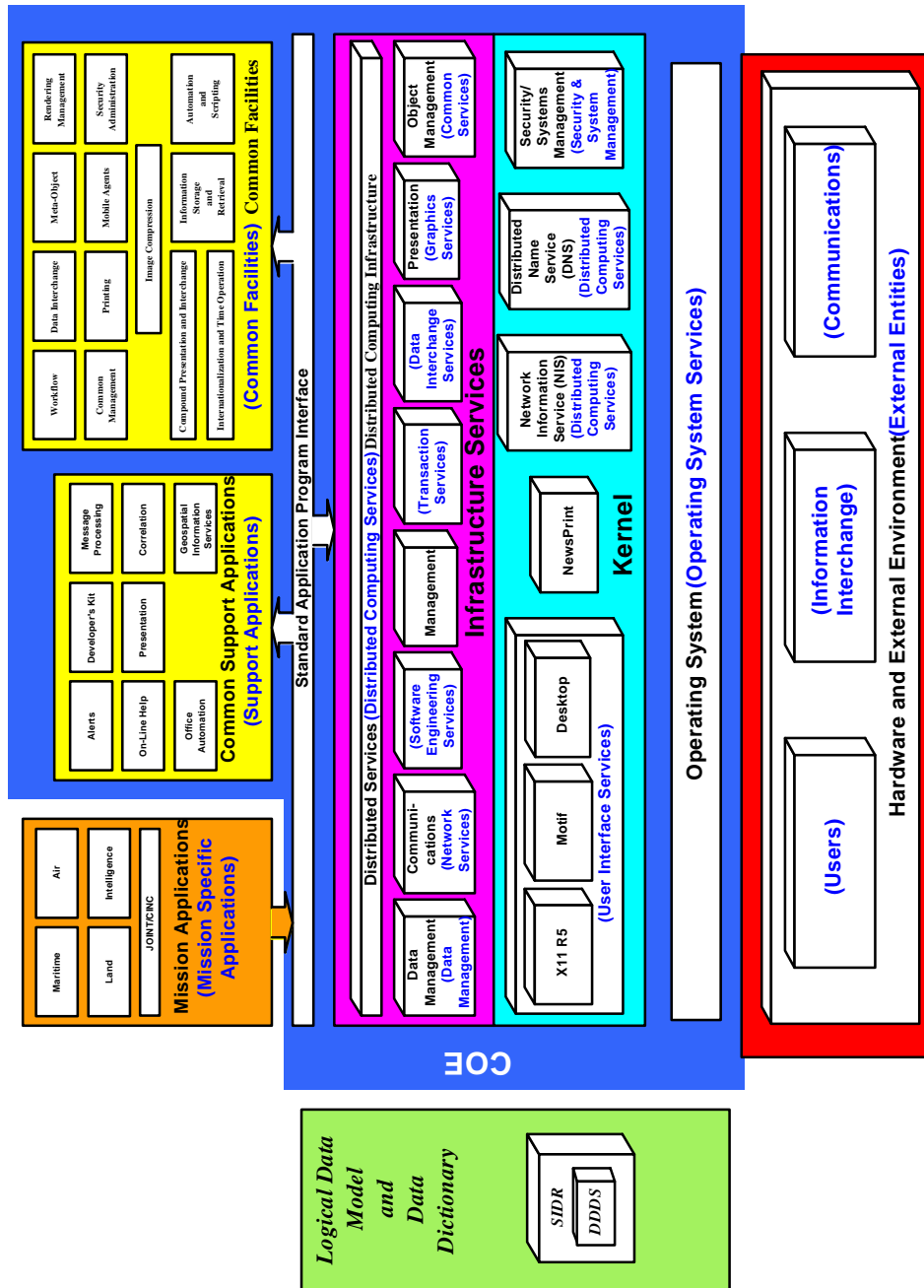


Figure 3-1. Merging of Intelligence Community Reference Model and CIIF Reference Model into the DII COE Reference Model



Note:

Names from the DII COE reference model look like this.

Names from the Draft Intelligence Community reference model look like this (and are braced by parentheses)

Names from the CIIF reference model look like this

Names from other sources look like this

Figure 3-2. Intelligence and CIIF Reference Models Merged with DII COE Reference Model

- The COE “Distributed Services and Object Management” infrastructure services have been divided into Distributed Services, shown as spanning all of the infrastructure services, and Object Management, shown as one of the infrastructure services
- The External Environment has been recast to show Users, Information Interchange, and Communications, which aligns more closely with the generic POSIX reference model and the Intelligence Community Reference Model
- A reminder that a data model is part of the architecture is shown along the left side, in lieu of the specific databases shown in the current COE Reference Model

To further illustrate the correspondence between the CIIF Reference Model and the merged COE reference model, Figure 3-3 expands the MCG&I and the Object Management components to show the particular services identified in the CIIF Reference Model. Relative to the DII COE, the CIIF is primarily a set of imagery services that should be added to the MCG&I service area. In addition, the CIIF Reference Model assumes the availability of certain basic object services, so these are identified as well.

Some parts of the CIIF may be viewed as specialized to the national imagery mission sufficiently to look like “applications”, i.e., components that obtain services (including CIIF services) from the DII COE but are not themselves part of the common MCG&I services. Some of the CIIF services may duplicate services already present in the DII COE and a closer analysis of each such service will be needed to determine which of several possible courses to take—omit it in favor of an existing DII COE service, replace a DII COE service with the CIIF service, or combine a CIIF service with a DII COE service to create an improved DII COE service. It is likely that many of the subordinate services needed by the CIIF are available in and should be obtained from the DII COE.

3.3 Architecture Analysis

The CIIF interface specifications represent a high-level architecture of USIGS imagery interoperability services. By using an interface specification language (ISO IDL) that supports inheritance, the interfaces can be made to reflect the hierarchical relations of the services to which they give access. For example, the CIIF Catalog Access Facility includes interfaces inherited from the more general Storage and Retrieval Facility.

The USIGS envisions a flexible distributed infrastructure in which standard interfaces are used to facilitate evolutionary upgrades and in which software duplication is minimized. The fact that CIIF interfaces are being specified in ISO IDL does not represent an intention on the part of the CIO to implement the CIIF in a CORBA environment. However, it does represent a recognition that a valuable approach to interoperability is the publication of standard interfaces that are independent of the implementations to be accessed through the interfaces.

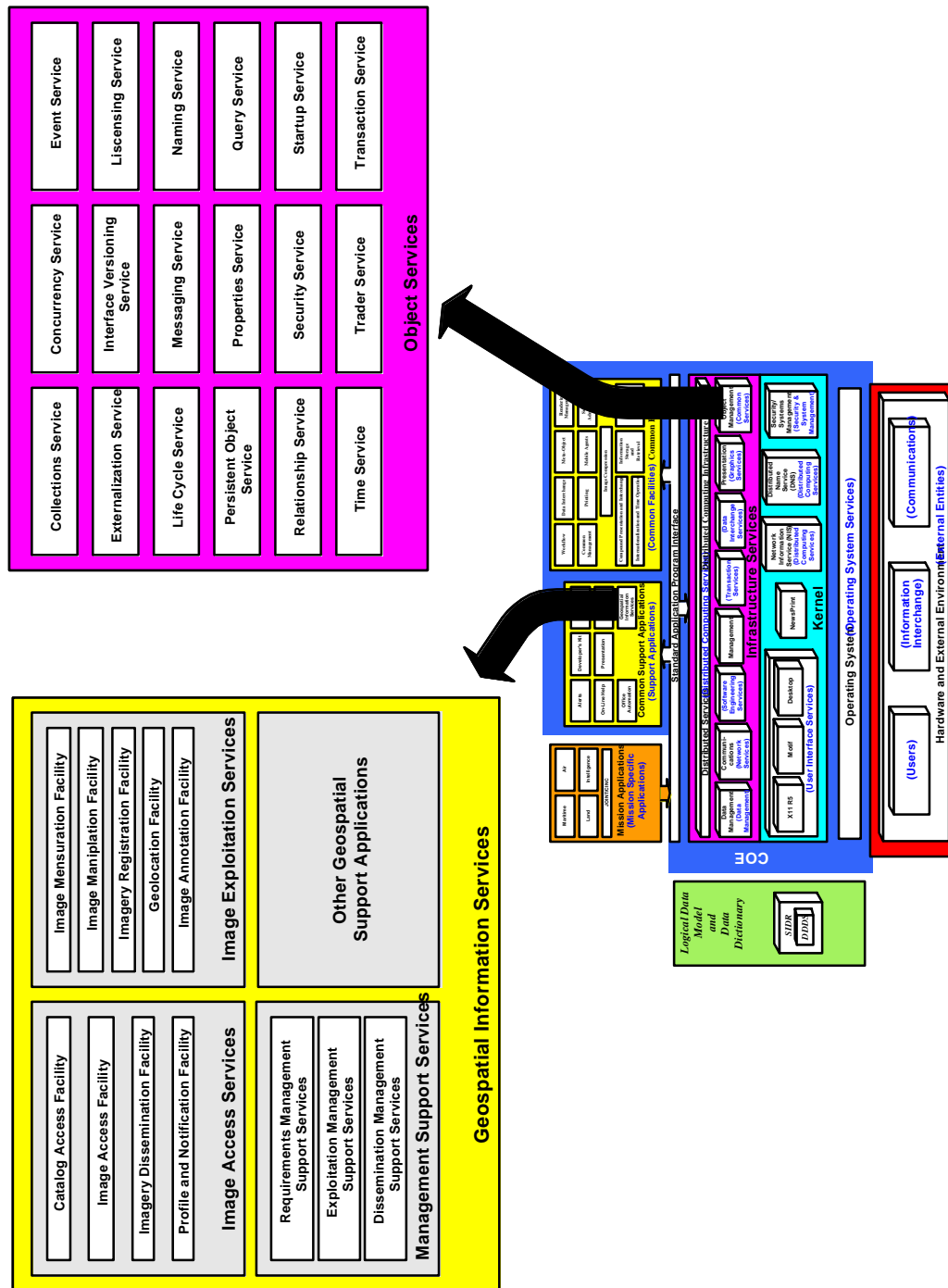


Figure 3-3. Detailed View of MCG&I and Object Management Services in the Merged DII COE Reference Model

The DII COE envisions a broker-based distributed environment. The DII COE will use DCE services as the principal distributed computing infrastructure in the near term, and add CORBA as that technology matures. DISA envisions that the DII COE will support both DCE and CORBA applications, and prefers that the CORBA technology duplicate DCE functions as little as possible. (See the CORBA excerpt from the Distributed Computing SRS in Appendix C.)

Generally there is similarity between DCE and CORBA. Interfaces are defined with IDL. IDL is maintained separately from the implementation code. Implementation entails the use of stubs generated by the IDL compiler. A server uses distributed infrastructure services to register with the broker so that clients can find it. Legacy code can be integrated by “wrapping” it in IDL.

The CIIF can be implemented over either DCE or CORBA, since both supply the necessary fundamental distributed computing services, and both support the specification of interfaces separately from implementations. So the CIIF and the DII COE architectures are generally compatible. However, implementors must provide for any inherited interfaces identified in the IDL for a facility, including those associated with CORBA Services and CORBA Facilities.

3.4 Interface Analysis

As described more fully in Section 2, a CIIF specification is a set of APIs and a description of sequencing semantics. The sequencing semantics describe the ways in which interfaces are related and conventions about how to use the interfaces and interpret parameters. However, no details or constraints are given on how the operations accessed through the interfaces should be implemented. A CIIF specification and the architecture that describes the interrelations among its services and other services of the USIGS is concerned primarily with the interfaces through which a user audience accesses the services that bear the interfaces.

The use of ISO IDL to specify CIIF interfaces means that the interfaces are organized around information objects, and that the specification is written according to a set of formal rules. Although the normal implementation course would be to compile the IDL into client and server stub modules in one of the programming languages for which an IDL binding has been defined, the IDL need not be used that way. In any event, even if stub modules are generated, they implement only the interface (the function name and parameters) specified in the IDL and an interface to the broker that will arrange connections over the distributed computing infrastructure between clients and servers. The functional code remains largely unconstrained. How it is organized and what supporting services it calls upon (e.g., DBMS) remain to be decided by the implementation team.

With respect to the DII COE, a CIIF service can be implemented to call upon DII COE services such as database access using SQL2 and desktop services using CDE while still bearing the prescribed CIIF interface to users.

Issues raised by considering CIIF interfaces in the context of the DII COE include the following:

- Style—the use of IDL for specifying CIIF interfaces is very different from the traditional API specification. There appears to be advantage to the CIIF approach, and the DII COE is likely to benefit in the long term by adopting it. The primary advantages of using IDL to specify interfaces are as follows:
 - Clarity and completeness—the formal status (draft international standard) of OMG IDL ensures unambiguous interpretation; the conventions of defining data types and exception conditions result in a more complete specification.
 - Architectural coherence—the preferred approach to developing a set of interfaces to be specified in IDL is to create an architecture that defines services for the data objects of interest, and then to create an interface architecture that gives access to the service architecture in a minimal but robust and complete way; through inheritance, the interface architecture can be made to reflect hierarchical and other relationships among the services that will bear the interfaces.
 - API stability and reusability—ISO IDL supports the inheritance of interfaces from one definition module to another; this is an elegant way to reduce the risk of redundant definitions and to represent the hierarchies of function likely to exist in a service architecture; it is also a mechanism for extending an API without changing the existing API.
- Duplication and overlap—CIIF services fall into several categories already part of the DII COE, among them data management, data interchange, imagery exploitation, system management, security administration, directory services, compound presentation and interchange; these areas lie in two of the major software layers of the DII COE—platform services and support applications.
- Inheritance of other interfaces—ISO IDL supports the inheritance of interfaces specified elsewhere. In considering the implementation of CIIF functions, it is necessary to understand whether an inherited interface is represented by an existing implementation, and whether the implementation is in the DII COE or overlaps one that is.

3.5 Function Analysis

In addition to defining interfaces, a CIIF specification describes usage conventions. The interfaces and usage conventions characterize the high-level functions that a CIIF service is expected to perform. However, a CIIF specification does not identify the detailed functions that the CIIF must implement or invoke to perform the services represented by the interfaces. From the point of view of a software developer, the CIIF is not a functional specification, and there is considerable freedom for designing an implementation.

On the other hand, although the DII COE can facilitate implementation by offering needed support functions, it constrains implementation options. This will be true to the degree that a CIIF implementation seeks to use DII COE services wherever possible to avoid duplication, even when a better rendering of a support service is deemed to exist than the one in the DII COE. However, an analysis of how well the DII COE supports CIIF implementations must be deferred until CIIF developers begin to define implementation details.

Although detailed analysis must be deferred, the high-level CIIF functions identified as interfaces by the IDL can be compared with functions and APIs being offered by the DII COE. Where there is overlap, more detailed analyses would be suggested to decide how the DII COE service can best support both the existing need and the CIIF. The area of greatest apparent overlap is MCG&I, which has been represented by the JMTK in the DII COE. It should be realized that the comparison performed for this report was being conducted at a high enough level of abstraction that what appears to be overlap may prove to represent quite different functions at some finer level of granularity.

The following tables compare CIIF services with DII COE services, including MCG&I services. Comparisons of APIs must be deferred to a more detailed stage of analysis, which is beyond the scope of this report. As discussed in section 3.1, it is likely that the services identified by both DII COE and CIIF have overlapping functionality. The alignments or “closeness of fit” that exists between the service areas indicate many areas for replacement, merging, or collaboration, as is suggested in the reference model of Figure 3-2.

Table 3-1 presents the correlation between the COE’s infrastructure services, operating system services, and common support applications except for those identified with the MCG&I support application. One can see from the symbols that functionally there is much commonality to the service areas. It is important to note that the CIIF document acknowledges the common facilities as broad area facilities that extend beyond the imagery, and now the MCG&I, community’s area of responsibility. Table 3-2 presents the service areas that are the CIIF’s primary focus and closely related to the DII COE’s MCG&I services.

As one compares the DII COE’s common support applications with the proposed reference model’s support application area there does seem to be a larger perspective where the common facilities cannot replace the support application itself. An example of this pertains to the first entry in table 3-1. The Office Automation support application is necessary for both reference models and is not replaced by the Automation and Scripting Common Facility or the Compound Presentation and Interchange Common Facility. However, to support the distributed object computing environment and take advantage of the inheritance concept, the common interfaces are necessary.

Infrastructure services are required for both reference model’s application platforms in much the same way. User interface services are also common to the platform service interfaces

as are those services resident within the Kernel such as Security/Systems Management and various distributed computing services.

Table 3-1. CIIF Common Facilities Mapped to DII COE




















CIIF Common Facilities	DII COE Service	Description	Alignment /Correlation
			   High Medium Low/None
Automation and Scripting Common Facility	Desktop (Kernel) Office Automation (Common Support Application)	This common facility defines conventions and interfaces acting on objects typically documents, paragraphs, spreadsheet cells, ...which expose enough of their attributes to be driven by scripts and macros.	 <p>The alignment is driven more by the subject of the function than the function itself. However, similar capabilities are behind the COE services and overlap with the <i>Compound Presentation and Interchange Common Facility</i>.</p>
Common Management Common Facility	Security/System Management (Kernel) Management (Infrastructure)	Provides utility interfaces for abstract functions such as control, monitoring, security management, configuration, and policies needed for system management operations (e.g., adding new users, setting permissions, installing software, ...).	 <p>The common management facility mirrors system administration and security management functions present within the two COE service areas.</p>
Compound Presentation and Interchange Common Facility	Desktop (Kernel) Office Automation (Common Support Application)	Supports compound documents that can be customized, used collaboratively, and available across multiple platforms. Includes the associated storage and interchange of data objects.	 <p>As stated above, there is overlapping functionality primarily in the creation, use, and presentation of compound documents.</p>
Data Interchange Common Facility	Data Management Services (Infrastructure) Presentation Services (Infrastructure)	Defines a common information model and information encoding within that model supporting character data, imagery, graphics, multimedia documents, electronic mail, and other sorts of data. enables objects to interoperate through exchange of data, and many kinds of data transfer (e.g., bulk data transfer, formatted data (TIFF, GIF, EPS, NITF... exchange), structured data, domain-specific objects, and objects-to-encapsulated software)).	 <p>Data Interchange functionality is performed throughout the COE but perhaps most strongly in the Infrastructure Services. Presentation Services, under Multimedia, lists formats for all media types, including imagery-NITFS, JPEG, and others. Data Management requires data exchange between all platforms, users, and databases in the DII.</p>
Information Storage and Retrieval Common Facility	Data Management Services (Infrastructure)	The Common Facility comprises the higher level specifications for distributed applications. Specifications will be relevant to a wide range of information services, including data base access and information highways.	 <p>Functionally, this common facility is a subset of the COE service area "Imagery Specific" functionality may fall under CIIF's IAF or CAF services.</p>
Imagery Compression Facility	Data Management -- Data Access (Infrastructure) - Compression	Provides standard interfaces to generalized services for imagery compression and decompression, and conversions between internal and standard formats.	 <p>As yet, neither system has specifically defined this area; there is potential for a more complete correlation.</p>
Internationalization and Time Operations Common Facility	Distributed Services & Object Management -DCE Time (Infrastructure)	Enables interfaces across languages and differing cultural conventions. Specifically, numeric and currency conventions and reconciling time zones.	 <p>There is some overlap in functionality; however the details and implementation differ.</p>

Table 3-1. CIIF Common Facilities Mapped to DII COE (concluded)

CIIF Common Facilities	DII COE Service	Description	Alignment /Correlation
			   High Medium Low/None
Meta-object Common Facility	Distributed Services & Object Management	Defines the interfaces and sequencing semantics needed to create, store, and manipulate object schemas that define the structure, meaning, and behavior of other objects	 DII COE just recently added object management and is currently defining this area.
Mobile Agents Common Facility		Ranges from interfaces necessary over massively distributed information systems to mobile information systems, intelligent workflow systems, and agile corporation information structures (INTRANETs).	 No COE equivalent.
Printing Common Facility	COE Printing Services	Defines coordinated set of facilities and standards ranging from printing one simple document to high volume production printing involving many documents, document types, printers and printer types.	 Functionally, the Common Facility encompasses a wider range of services targeted at the modern distributed (commercial) office.
Rendering Management Common Facility	Communications Services (Infrastructure) Windowing Service (Kernel)	The common facility is primarily concerned with user interface (input/output) media such as screens, printers, plotters, scanners, sound, speech, camera, mouse, keyboard, and security devices.	 Rendering management is primarily associated with user interface functions but does have some applicability to the two named DII COE service areas.
Security Administration Common Facility	Management (Infrastructure) Security /Systems Management (Kernel)	Defines interfaces and control mechanisms to facilitate security protections such as user and resource registration; password, permissions, and audit trail maintenance; discretionary, mandatory, authentication, and key management access control; and security classification downgrading. There is a very close alignment here.	 Depending on the specific operation, the common facility's functions may apply to either or both COE service areas. The requirements are very well defined and virtually an exact fit.
Workflow Common Facility	Management (Infrastructure) Presentation (Infrastructure)	Defines management and coordination of objects/components that are part of a work process. The processes can be production-based, pre-defined by policies and procedures, ad-hoc, and coordination-based.	 Recent changes to the COE reference model relocated several workflow management type functions into the Management and Presentation Infrastructure services.

As table 3-2 points out there is a wide variance of correlation with MCG&I support applications and services identified by the CIIF for imagery oriented applications. The table depicts many which are common across diverse mission applications as well as those which may be specific to the intelligence community. Future development will also refocus the CIIF effort to the wider MCG&I community under NIMA management.

3.6 Compatibility with DII COE Data Architecture and Standards

The DII COE places strong emphasis on shared data. It includes not only the use of shared data management facilities, but also shared data definitions and databases. The current MCG&I domain in the DII COE defines data formats and data storage and access functions.

Table 3-2. CIIF Imagery Services Mapped to DII COE MCG&I Services






















CIIF	DII COE Service	Description	Alignment /Correlation
			   High Medium Low/None
Image Manipulation Facility	Image Manipulation Services (MCG&I Support Application)	A subset of imagery exploitation services such as roam, zoom, rotation, orientation, image resampling, edge sharpening/smoothing, brightness and contrast, color table manipulation and pseudocolor assignment	 A very close alignment of CIIF to COE.
Image Mensuration Facility	Mensuration Services (MCG&I Support Application)	Geometric measurements from monoscopic and stereoscopic imagery providing linear, curvilinear, and multi-dimensional measurements and summation services capable of supporting conversion and feature orientation requests.	 A very close alignment of CIIF to COE.
Image Registration Facility	Registration Services (MCG&I Support Application)	Performing spatial correlations on the basis of image or graphic content. Aligning, co-registering, fusing, warping, rectifying images and/or graphics such as image to image, image to geospatial, image to graphics product creation services (e.g., mosaicking, radiometric matching, image combining, overlays).	 Closely aligned; however, CIIF interfaces may need more capability than COE can economically provide.
Geolocation Facility	Location Services (MCG&I Support Application)	Performing point location functions consisting of Rapid Positioning Capability, geolocation, grid, imagery detail, and coordinate conversion services.	 The functionality required by CIIF or COE is virtually indistinguishable.
Image Annotation Facility	Annotation Services (MCG&I Support Application)	Provides capability to place symbology, text, and graphics integrated with imagery to highlight significant content.	 The functionality required by CIIF or COE is virtually indistinguishable
Catalog Access Facility (CAF)	Data Management (Infrastructure)	Provides interface for client to imagery library catalog services. Product discovery, metadata and database resource access, indexing, and directory maintenance are services enabling imagery queries and browsing as well as catalog updates.	 The CAF is currently imagery specific; COE does not provide a geospatial query functionality.
Image Access Facility (IAF)	Data Management -- Data Access Services (Infrastructure)	Defines standardized methods for storing and retrieving imagery and imagery-related data within shared (distributed) libraries.	 No current correlation with COE except for the planned flexibility leading to a convergence of character-based applications with digital photographic, geographic, and drawing applications consistent with commercial software trends.
Profile and Notification Facility	Data Access Services - Standing Request	Interfaces defining local and global interest profiles which establish criteria for automatic notification, retrieval and delivery of relevant imagery, imagery products, or meta-data.	 Further investigation is necessary to determine DII COE geospatial support.
Image Dissemination Facility		Enables the formatting, delivery, routing, and prioritizing of imagery (products) and the tasks associated with product distribution.	 No COE equivalent.
Automatic Target Recognition Candidate Service Area		Provides tools which automatically detect, categorize, count, and determine relationships between objects in imagery.	 No COE equivalent.

Table 3-2. CIIF Imagery Services Mapped to DII COE MCG&I Services (concluded)

CIIF	DII COE Service	Description	Alignment /Correlation
			   High Medium Low/None
Image Synthesis Candidate Service Area	Elevation and Terrain Services (MCG&I Support Application)	Object modeling, synthetic image generation, image perspective transformation (IPT), and 3D fly-by generation are currently very "government-only" oriented services.	 The IPT service does appear in COE's MCG&I (Elevation and Terrain) services
Image Understanding Candidate Service Area	Data Comparison Services Elevation and Terrain Services (MCG&I Support Application)	Applies knowledge-based inference techniques to extract intelligence from imagery beyond factual scene content.. Change detection, pattern and object recognition, feature extraction, and terrain analysis are proposed services	 Change detection is common to the CIIF facility and Data Comparison services. The feature extraction and terrain analysis services descriptions indicate common functionality.
Requirements Management Candidate Service Area		Provides imagery collection nomination and feedback status services. This includes resource information and prioritization and collection planning capabilities.	 No COE equivalent.
Exploitation Management Candidate Service Area		Provides exploitation task assignment, status, and resource information services.	 No COE equivalent.
Dissemination Management Candidate Service Area		Interfaces defining resource availability, performance criteria and status, and imagery storage, retrieval, and delivery (strategy) services.	 No COE equivalent.

Although this is a very important topic for interoperability, it is substantially beyond the scope of the present analysis. One reason is that it is a large subject. There are a number of repositories of image data, both government and commercial. There are many formats and database mechanisms associated with those repositories. A second reason is that the USIGS data architecture has received little attention to date, and the implementations of CIIF data access functions have not been designed, so it is too soon to carry the data analysis very far.

The initial CIIF specification includes discussion of data formats (e.g., NITFS), but does not include details about database management systems. However, many of those details for national imagery data are addressed in the specifications for the National Imagery Library (NIL), Command Imagery Libraries (CILs), and the Imagery Product Libraries (IPLs), and the Image Access Services CIIF are intended to be used to access those repositories. There is a need to understand the degree to which data required, originated, disseminated, and stored by USIGS are already part of existing databases in the DII COE—through the JMTK, for example. In any case, there is a need to decide how USIGS data should be organized and

managed within the DII COE. It seems appropriate to perform this analysis at the same time as the detailed comparison of CIIF APIs with JMTK APIs and other DII COE APIs.

3.7 Packaging for Integration into the DII

The organization of the CIIF into DII COE segments using DII COE naming and directory conventions should not present significant difficulty for CIIF developers. However, the implications of organizing CIIF software as DII COE services packaged as DII COE segments should be explored further. When prototype implementations of CIIF interfaces are available, it will be appropriate to identify the steps needed to package them as DII COE segments at Level 5 or higher interoperability compliance (see Table 2-3). Other facets of the analysis would examine the extent to which CIIF services can be built over DII COE services to minimize duplication of function, and would identify the additional steps needed enable CIIF to achieve Level 8 compliance.

3.8 DII COE Architecture and Technology Processes

As the agency responsible for definition and management of the DII COE, DISA has published guidance and established procedures and organizations. The guidance defines functional and technology objectives. The procedures and organizations facilitate activities through which DII COE user agencies can define requirements, agencies and vendors can offer technology, and developers can implement DII COE services. The CIIF represents both services required for national imagery processing and facilities that require infrastructure services from the DII COE. CIIF integration with the DII COE will entail both an expression of support requirements from the intelligence community and an analysis with DISA of how and where CIIF services should be added to the DII COE. (Figure 3-2 is an initial indication.)

3.8.1 The Architecture Oversight Group (AOG) and Technical Working Groups

The purpose of the AOG [21] is to identify, document, and validate DII COE requirements. These requirements are defined as COE tools, and services and capabilities needed by the warfighter to understand the impact of environment on both friendly and threat information and on weapons systems and contingencies in a theater of operations. The AOG is chaired by the DII Chief Engineer, DISA, Center for Computer Systems Engineering, and is composed of representatives from the Services and DoD Agencies.

Each Service and Agency designates one or more technical representatives to participate in the DII COE AOG. A representative is responsible for soliciting Service or Agency input on technical issues addressed by the group, for coordinating a Service or Agency position with regard to these issues, and for articulating that position when Service and Agency consensus on working group actions is required.

The AOG has the following responsibilities:

- a) Identify, document, and validate COE requirements on behalf of the Services and Agencies.
- b) Identify the Service and Agencies' existing systems, models and data bases that can be employed or interfaced to satisfy DII COE requirements.
- c) Define the policies, procedures and ADP support requirements in collaboration with member Services and Agencies.
- d) Monitor COE requirements and definition, migration, and development or enhancement of specific DII COE capabilities to ensure satisfaction of COE requirements.
- e) Establish, as required, appropriate Technical Working Groups (TWGs), for the detailed functional definition, review, coordination, clarification, refinement, and fielding of DII COE capabilities and services.
- f) Identify and request through appropriate channels those resources required to accomplish these responsibilities.
- g) Establish and maintain liaison with other working groups to ensure that changes to procedures and ADP systems are synchronized; that information between functional systems can be exchanged; and that applications warranting integration into DII COE are identified and incorporated.

The AOG is established as the DII COE Working Group for the COE Architecture. Technical Working Groups may also be created by the AOG to address other technical areas relating to DII architecture, integration, and engineering. TWGs provide recommendations to the AOG on issues relating to DII architecture and implementation in the functional or technical area addressed by the group. These recommendations are advisory in nature and reflect the combined input of the Service and Agency representatives participating in the group. The AOG considers TWG recommendations in light of the DII COE development strategy and program plan and directs implementation as appropriate.

3.8.2 Integration and Runtime Specifications (I&RTS)

The DISA Joint Interoperability and Engineering Organization (JIEO) has published the I&RTS to guide developers on how software intended to operate as a service within the DII COE or as an application that uses DII COE services should be organized and tested. The objective is to achieve a reliable, tool-assisted system for maintaining the DII COE and disseminating upgrades to user sites. Compliance categories were presented in Tables 2-3 and 2-4.

3.8.3 Analysis and Observations

Ultimately, USIGS and CIIF infrastructure *support requirements* should find expression in the DII COE SRS. This means that they should be presented to the appropriate TWGs for

analysis and comparison with similar requirements from other user organizations, so that the AOG receives an SRS that represents the entire user base.

On the other hand, CIIF services being proposed for integration into the service offerings of the DII COE should be presented to the appropriate TWGs as potential *implementations or architectures for implementation*. Again, the purpose is to put the CIIF proposals into the DISA process for identifying, evaluating, and adopting implementations. To the degree that implementations are being proposed, they should be packaged as DII COE segments as prescribed in the I&RTS.

Section 4

Issues and Opportunities

In this section the baselines described in Section 2 and the analytical observations of Section 3 are used to form conclusions about how the CIIF can be implemented as DII COE services. Also, where appropriate, suggestions are made about how the DII COE might be changed to advantage.

4.1 Technology Trends

The best course for integrating CIIF services with the DII COE depends on how the technology and product markets evolve over the next three years. Some of the trends most likely to have significant effect are summarized in this section.

4.1.1 Brokered Distributed Architectures: CORBA, DCE, and DCOM

A client-broker-server architecture has been adopted for the DII COE to achieve robustness, transparency, scalability, and evolvability. DCE is being deployed currently as the initial set of broker-based distributed computing services. CORBA is planned for addition within the next year.

DCE IDL is converging to CORBA IDL, and the CORBA IDL is a draft ISO standard. Principles are similar: interfaces are defined and managed separately from implementation code; interface definitions can be inherited; stubs in a chosen programming language are generated to connect clients and servers through the broker-based distributed infrastructure. Both C and C++ programming languages are being accommodated.

Distributed architectures whose broker mechanisms are based on ORBs are clearly the architectures of the future. Compared with DCE-based brokers, ORBs are more comprehensive, based on more powerful mechanisms, and represent a higher level of abstraction. The readiness of CORBA to fulfill the broker role of the DII COE depends primarily on the commercialization of essential CORBA Services and CORBA Facilities. That is proceeding, but is likely to require another year or two to reach a satisfactory state.

In the meantime, DCE contains security, directory, and broker mechanisms today that interoperate over virtually all commercially available computing platforms, including those in the DII COE. DCE answers many DII COE distributed computing requirements; without them, the DII COE would have to defer further some of its key objectives as a common infrastructure.

DCOM, a proprietary technology from Microsoft, is a distributed object technology that will be widely used in NT servers and workstations. It has components similar to CORBA—an object request broker, an interface definition language, and basic services that include

directory and naming, security, events, and many others. A large library of Microsoft Foundation Classes gives developers a more mature set of basic services than is available for CORBA developers today.

Interoperability between DCOM and CORBA environments is offered by several CORBA vendors. OMG, of which Microsoft is a member, is developing a specification for CORBA-DCOM interoperability, but that effort has slowed after a period of early activity in 1995. OMG is also forming a group to address DCE-to-CORBA transition issues.

4.1.2 Data Management

Currently there are separate standards for relational and object-oriented databases. However, both sets of standards are being revised for a next release. SQL3, the successor to the current relational standard (SQL2 or SQL-92), is on the way. In addition to relational standards, SQL3 will include standards for stored procedures along with multimedia and object-oriented data access standards. The SQL3 effort and the revisions to Object Database Management Group (ODMG) standard ODMG-93 are being coordinated to ensure close or complete alignment of object-oriented data access facilities in the two standards. The revised standards are expected in 1997. The OMG has adopted ODMG specifications for the most part to define the database aspects of the Persistent Objects Service.

4.1.3 Web Browsers

A web browser is a graphical user interface (GUI) to the World-Wide Web, in which sites and documents are related through hyperlinks. Netscape and Explorer are the market-leading products. Web browsers have matured to the point that soon they will be candidates to serve as the primary desktop application, giving access to local and networked resources as if they all were local. (CDE and Windows serve that role in the current DII COE, which includes Netscape as a Web access application.) Built-in security is likely within the next year through DCE Web Security and secure IIOP, among other technologies. (See also Section 4.1.5.)

Netscape has announced plans to incorporate a light-weight CORBA-compliant ORB into its browser by early 1997. Since Netscape is on most DII COE desktops, this implies that most DII COE user systems can have an ORB simply by upgrading to that version of Netscape when it is available.

4.1.4 Mobile Code

The Java programming language and execution environment from Sunsoft have popularized the idea of “applets”. An applet is a small application module that can be downloaded and executed by a Web browser when a Web page script contains the appropriate HTML command and the browser platform includes an execution environment for the applets. The technology gives great freedom to developers to enhance the browser interface with

custom, interactive functions. Viewed another way, developers may present their applications through the familiar and ubiquitous browser interface. As a programming language, Java is similar to C++. Java source code is compiled into an intermediate code that is then interpreted on the destination platform. Java chips have been announced, as have full Java compilers, to improve execution speeds on the browser platform.

ActiveX from Microsoft is a similar facility in which OLE objects can be downloaded (in binary form) and executed within the Explorer web browser. Explorer also can execute Java applets. Future releases of Netscape will be able to execute ActiveX applets as well as Java.

4.1.5 Security

A subject of increasing commercial importance for distributed computing is security. The first requirement is that it exist. Another is that, as much as possible, it be transparent. Many products and product plans have been announced in the past year. Through appropriate mechanisms for defining and managing access permissions, encryption levels, and auditing, the intent is that information and processing resources can be protected to any degree desired. Then, for a user who has identified himself to a computing environment upon initial entry, all resources and information are accessible to the levels for which he has permission; conversely, every resource is protected to an appropriate level from every user and from various other threats.

The idea of multilevel security has been under study and development in the DoD and the intelligence community for several years. The goal is to enable a single workstation through a single user interface to access two or more environments that traditionally have been secured at different levels through physical isolation. Current practice is to preserve the secure enclaves and to control access from one enclave to another with a software mechanism called a "guard". In principle, a flexible, robust set of security functions can make guards and enclaves unnecessary. The goal of multilevel security is to replace physical boundaries with virtual boundaries.

The DCE security service, which is based on the Kerberos system, achieves many of these objectives using secret keys. A DII COE requirement for public key encryption cannot be met by the current DCE technology, but the definition of public key facilities for DCE is under development.

The OMG has adopted a security specification for processes that interoperate through a single ORB. Commercial implementations will appear in the next year. An additional specification to address processes that interoperate through interoperating ORBs is in preparation.

The Open Group (formed by the merger of X/Open and OSF) has defined both short term approaches to security in an open, distributed system and a long term program to evolve standards. Their white papers and technology programs constitute a practical, well informed

view of the state of the art and how it can evolve to answer the needs of both commercial and government organizations for unobtrusive, robust security in open, distributed environments.

Security mechanisms are part of the DII COE, and will be evolved toward the multilevel ideal as technology permits.

4.2 Reference Models

The draft Intelligence Community Reference Model (Figure 2-4) is a rendering of the TAFIM technical reference model in which distributed computing services are shown as explicit platform components. A reference model derived by combining the Intelligence Community Reference Model and the DII COE Reference Model, shown in Figure 3-2, seems to depict the essential features of each of the separate models. It includes a grouping of services into Kernel, Infrastructure, Common Facilities, and (Common) Support Applications, which is more closely aligned with the Object Management Architecture of the OMG without disturbing the key groupings of the DII COE.

A further rationalization of Figure 3-2 would remove product references from the kernel and combine certain function groups into larger categories of service. Figure 4-1 shows such a model. This model also removes the distinction between kernel and infrastructure services by considering both to be infrastructure.⁴ The idea is to present a DII COE reference model that is more closely aligned with the levels of abstraction in the POSIX, TAFIM, and JTA reference models on one hand, and that reflects the categories of application software found in the Intelligence Community (draft), OMA, and CIIF reference models on the other. Since the DII COE has already adopted DCE as a distributed computing technology and is planning to introduce CORBA, it is suggested that DISA and the DII COE community would be well served by adopting the refinements introduced in the combined Intelligence Community/DII COE reference model.

4.3 APIs

The CIIF is based on an architectural approach in which interfaces are defined before implementations. Further, they are defined in ISO IDL, which enables the hierarchical nature of a family of APIs and the services that bear them to be represented through inheritance relationships. The DII COE APIs are either de jure standards for which commercial products exist or they are interfaces defined for particular legacy products and service implementations that have been adopted from one of the DII COE user organizations.

⁴ It is acknowledged that the ideas of a kernel segment and a product-based reference implementation are important for deploying the DII COE. A diagram that represents those aspects will continue to be useful in that context. The kernel might contain services from each of the infrastructure service groups shown in Figure 4-1.

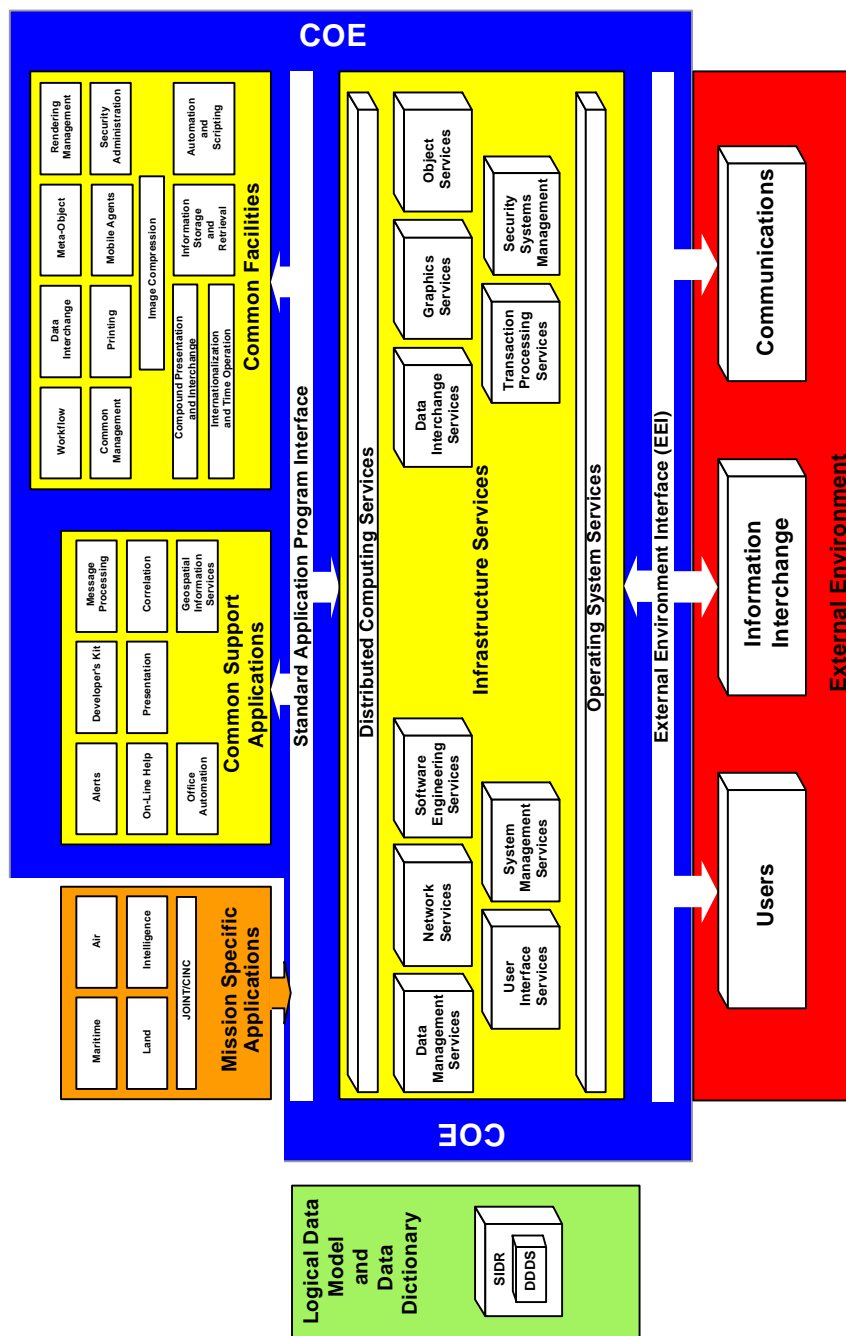


Figure 4-1. Rationalized DII COE Reference Model

The DII COE technology adoption and API configuration control processes have been driven by the need to field a comprehensive set of operational services in the near term. The press of time in those circumstances has not seemed to accommodate the CIIF approach of defining an API architecture for services independent of their implementation.

However, now that an initial DII COE has been fielded, the planning for its evolution would seem to be well served by adopting the CIIF approach. As the responsible agency for the DII COE, DISA would seem to have two principal concerns for the long term:

- Stability and extensibility of the APIs
- Evolution and integration of the APIs and of the infrastructure

The CIIF approach includes two process steps that support these objectives. The first step is to “mine” current applications and services to identify the high-level functions that must be made available. The second step is to define a service architecture and document its structure and its APIs with ISO IDL. The mining step ensures that the service architecture is as complete as it can be—by viewing existing services as expressions of earlier requirements assessments. In the case of the DII COE, this includes the idea that the needs of all branches of the military are included. (The AOG processes described in Section 3.8 could adopt the “mining” idea as a particular technique for requirements analysis.) At the same time, the use of IDL to define the APIs means that the initial API can be retained even when additions or modifications are needed, because the inheritance mechanism enables extensions from the existing base. Applications already based on the API are not affected when extensions are made.

4.4 Standards Compliance

The CIIF should comply with standards that have been identified in the JTA, the TAFIM-compliant document that identifies the standards foundation for the DII COE. Where this is not possible, CIIF applications should be selected that comply with international or national standards or, failing that, applications that comply with publicly available specifications for interfaces (APIs) that have significant acceptance in the marketplace.

For CIIF applications that do not conform to standards included in the JTA, but that do conform to national or international standards, it will be necessary to propose them for inclusion in the JTA and to encourage the adoption of their interface specifications by industry consortia and standards organizations. For those that comply with published specifications that are not specifications for international or national standards, it will be necessary to ensure that they are documented for use in the DII COE.

4.5 Data Management

The NIL and the CILs are the proposed principal repositories for national imagery. The Image Access Services Specification [22] defines the APIs through which the NIL and CILs

would be accessed. Each of these technologies—the libraries and the access services and APIs—should be added to the DII COE. The data model for the libraries should be designed to incorporate standard data names and definitions where possible. The access services should be designed to integrate with available DII COE data management services and APIs. Finally, the CIIF image access APIs should be added to or otherwise integrated with the DII COE data management APIs.

For the present, the access of imagery in the DII COE will be through the proprietary mechanisms of DBMSs that are part of the current configuration of the DII COE. Will the DBMSs and other data management components of the DII COE be migrated to operate over DCE and then CORBA? Presumably that is intended in the longer term, but does not appear to be in the offing anytime soon. Until it does happen, applications will have to use different interoperability mechanisms for data access and for other distributed functions.

The DCE distributed file system (DFS) is being deployed gradually in the DII COE and may eventually replace other distributed file systems. Among its advantages are an automatic cache management facility to reduce network traffic and increase responsiveness, and integration with the directory, security, and time services of DCE. Those facilities may be useful in the initial implementation of the IAS CIIF.

4.6 Distributed Computing

DCE is being installed in the DII COE, and CORBA is planned for the next year. Microsoft Windows and NT are being accommodated. According to the JTA, DCE RPC is mandatory if RPC services are needed, and CORBA is mandatory if distributed object services are needed.

The list of CORBA Services and CORBA Facilities given in the draft DII COE Distributed Computing SRS should be expanded to include those needed for the CIIF, and phased to match the timing of the CIIF.

If a developer wants to create a service that registers with a DII COE broker or wants to create a client that sends requests to a DII COE broker, should he write for DCE or CORBA? This is one of the biggest questions the DII COE faces: how to move to DCE, then add CORBA. To take full advantage of DCE, as many legacy services and applications as possible should be adapted to DCE. Is there an elegant way to accommodate DCE and then add CORBA so that applications, having once been recast to use DCE, can then be changed one more time to exploit CORBA? One idea is that most of the effort lies in restructuring the legacy code into distributable components; the subsequent step of “wrapping” these components for interoperation through DCE or CORBA may be small enough that it can be done twice if necessary. The best course is likely to depend on the particular application or service and the rate at which other applications and services become available through DCE and CORBA mechanisms.

The Catalog Access Facility of the CIIF is concerned with locating imagery data in response to queries. The directory services of the DII COE should enable much of the CIIF Catalog facility to be couched in logical rather than location-specific terms. Similarly, the CIIF Image Access Facility should be able to use the logical description of an imagery source obtained from the Catalog facility to find the physical location transparently through DII COE directory facilities. Both the data management and the distributed computing services of the DII COE will include directory functions.

4.7 Mapping, Charting, Geodesy, and Imagery (MCG&I)

Currently these functions are represented in the DII COE by the JMTK, which emphasizes mapping, charting, and geodesy. The MIG project is intended to accelerate the integration of imagery services into the DII COE. The attempt to implement the JMTK by combining legacy applications from each of the services has encountered obstacles that are preventing its completion in time for the next release of the DII COE. In the longer term, the realization of the JMTK Objective Architecture should alleviate these difficulties.

For the present, there is a need to review the functions and architecture of the JMTK and the MIG proposal to ensure that the needs of the CIIF are being addressed. Changes should be recommended where appropriate. The concerns include not only functional coverage but architecture and implementation plans.

4.8 DoD Acquisition and Standards Reform

On 29 June 1994, Secretary of Defense William J. Perry's memorandum, "Specifications and Standards—A New Way of Doing Business" [23], directed the military departments and other DoD agencies to "...use performance and commercial specifications and standards in lieu of military specifications and standards, unless no practical alternative exists to meet the user's needs." This directive, combined with increasingly effective methods by industry consortia to achieve consensus technology standards through public processes opens up practical avenues for Government agencies to ensure that their requirements are reflected in industry specifications and standards. In this climate, the agencies responsible for the USIGS and the DII COE should be participating or planning to participate in the OMG, the Open GIS Consortium, and the Open Group, among others.

4.9 Conclusions

The DII COE is intended to provide a DoD infrastructure capable of supporting all widely used information services required by mission applications, including intelligence applications. The architecture is client-broker-server, with DCE as the near term technology and CORBA in the longer term. Interfaces based on open standards enable components to interoperate and the architecture to evolve incrementally. They facilitate application portability, software reuse, and

plug-and-play system configuration. **The CIIF architecture is compatible with the planned DII COE architecture.**

The CIIF services of the USIGS Architecture require a number of supporting services from the DII COE, among them data management, security, system management, and broker-based distributed computing. **The integration of DII COE services with the underlying distributed infrastructure is only partial in the present early stage of the DII COE. It is not clear when DII COE services will be available through a common distributed computing infrastructure (DCE or CORBA).**

There appears to be overlap between the JMTK-based MCG&I services of the DII COE and the exploitation services of the USIGS CIIF. (It should be anticipated that some of the functions that appear to belong to a similar class may nonetheless be distinct when looked at more closely.) The overlap should be examined in depth to ensure that the MCG&I services in the DII COE incorporate both JMTK and CIIF requirements in the most appropriate way. It is known that the JMTK developers have concentrated on the MCG elements of MCG&I, with the expectation that the I(magery) element would be further specified by the national imagery community.

There appears to be overlap between the Image Access and Catalog Access CIIF and the data management facilities of the DII COE. As with MCG&I, an apparent overlap may resolve to distinct and justifiable variants when scrutinized more closely. The overlap should be examined in greater depth by exploring such questions as:

- Does the JMTK use the data management services of the DII to store and retrieve MCG&I objects?
- Does the Storage and Retrieval interface specified for the Catalog Access and Image Access CIIF constitute a more generic service that should become part of the DII data management facility?
- Should the DII COE support both low-level and high-level APIs for data services?

4.10 Action Plan

Several near term actions are needed to clarify further the best way to integrate CIIF services with the DII COE. Those actions are as follows, grouped by primary agent:

4.10.1 Proposed Actions for NIMA

- Update the CIIF reference model to reflect the findings of this report.

4.10.2 Proposed Actions for DISA

- Adopt the merged reference model of Figure 4-1 for the DII COE as a more useful portrayal of the role of distributed object services, common facilities, and standardized interfaces.
- Mandate ISO IDL (ISO/IEC DIS 14750) as the preferred language for defining DII COE interfaces. ISO IDL facilitates the organization of service interfaces into an architecture, and enables a precise, unambiguous, uniform definition of interface functions, parameters, and exception conditions. It is a mechanism for stabilizing APIs without prohibiting their extension and refinement.

4.10.3 Proposed Actions for NIMA and DISA

- Promote standards-based rather than product-based APIs.
- Formalize co-participation between the DII COE AOG and the Imagery Systems Management Committee (ISMC) and their respective working groups.
- Analyze relations between CIIF services and DII COE data management services and plan how they should evolve.
- Add DII COE APIs for common facilities, common support applications, and infrastructure services to the TAFIM and JTA.
- Participate in an ongoing, active Government partnership with industry in standards development.

List of References

1. Central Imagery Office, 12 April 1996, *USIS Objective Architecture Definition and Evolution*, CIO-2003, U.S. Department of Defense, Washington, D.C.
2. Defense Information Systems Agency Joint Interoperability and Engineering Organization, 26 April 1996, *DII Master Plan (Executive Summary)*, U.S. Department of Defense, Washington, D.C.
3. _____, 15 December 1995, *Architectural Design Document for the Global Command and Control System (GCCS) Common Operating Environment (COE)*, U.S. Department of Defense, Washington, D.C.
4. Department of Defense, February 1995, *DoD Intelligence Information System (DoDIIS) Profile of the Technical Reference Model.*, U.S. Department of Defense, Washington, D.C.
5. _____, 30 July 1994, *DoDIIS Client Server Environment (CSE) Integration Compliance Specification*, U.S. Department of Defense, Washington, D.C.
6. Assistant Secretary of Defense, 30 March 1995, *Technical Architecture Framework for Information Management (TAFIM)*, Volumes 1-8, Version 2.0, U.S. Department of Defense, Washington, D.C.
7. Defense Information Systems Agency Center for Standards, 30 September 1995, *Department of Defense Technical Architecture Framework for Information Management*, Volumes 1-8, Version 3.0 draft, U.S. Department of Defense, Washington, D.C.
8. Institute of Electrical and Electronic Engineers, October 1994, *PI003.0, Guide to the POSIX Open System Environment (Draft 17)*, IEEE Standards Board, Piscataway, New Jersey
9. Defense Information Systems Agency Joint Interoperability and Engineering Organization, , 22 August 1996, *Department of Defense Joint Technical Architecture*, Version 1.0, U.S. Department of Defense, Washington, D.C.
10. _____, 23 October 1995, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS)*, Version 2.0 preliminary, U.S. Department of Defense, Washington, D.C.
11. _____, July 1996, *DII COE System Requirements Specification (SRS)*, draft, U.S. Department of Defense, Washington, D.C.
12. _____, 29 September 1995, *GCCS Version 3.0 DCE Implementation Plan*, U.S. Department of Defense, Washington, D.C.

13. _____, 25 September 1995, *[GCCS] CORBA Migration Strategy Document*, draft, U.S. Department of Defense, Washington, D.C.
14. _____, 28 June 1996, *DII COE Version 2.0 (Series) Baseline Specifications*, U.S. Department of Defense, Washington, D.C.
15. Defense Mapping Agency, 20 July 1995, *Priorities for the Joint Mapping Toolkit (JMTK) of the GCCS COE*, U.S. Department of Defense, Washington, D.C.
16. Naval Command, Control and Ocean Surveillance Center (NCCOSC), 26 August 1996, "GCCS MIG: Integration of Imagery Capabilities in Support of C4I", briefing by Mark Kuzma, NCCOSC In Service Engineering Division (NISE)
17. Central Imagery Office, 8 May 1996, *Common Imagery Interoperability Facilities Reference Model (CIIF RM)*, Version 1.0, U.S. Department of Defense, Washington, D.C.
18. ISO/IEC DIS 14750, *Information technology -- Open Distributed Processing -- Interface Definition Language*
19. Central Imagery Office, 8 December 1995, *USIS Technical Architecture Requirements Document*, CIO-2004, U.S. Department of Defense, Washington, D.C.
20. Central Imagery Office, 13 October 1995, updated 31 May 1996, *USIS Standards and Guidelines*, CIO-2008, Version 1, U.S. Department of Defense, Washington, D.C.
21. Defense Information Systems Agency, July 1996, "Charter for the Defense Information Infrastructure (DII) Common Operating Environment (COE) Architecture Oversight Group (AOG)", U.S. Department of Defense, Washington, D.C.
22. Central Imagery Office, 24 May 1996, *Image Access Services Specification (IASS)*, Version 1.0, U.S. Department of Defense, Washington, D.C.
23. Perry, William J., 29 June 1994, "Specifications and Standards—A New Way of Doing Business", Memorandum from the Secretary of Defense, Washington, DC.

Bibliography

Books

- Ben-Natan, Ron, 1995, *CORBA, A Guide to the Common Object Request Broker Architecture*, McGraw Hill, New York, NY.
- Edwards, Jeri, Dan Harkey, and Robert Orfali, 1996, *The Essential Distributed Objects Survival Guide*, John Wiley and Sons, Inc., New York, NY.
- Mowbray, Thomas J., and Ron Zahavi, 1995, *The Essential CORBA: Systems Integration Using Distributed Objects*, John Wiley and Sons, Inc., New York, NY.
- Open Software Foundation, 1992, *Introduction to OSF DCE*, Prentice Hall, Englewood Cliffs, New Jersey.
- _____, 1995, *OSF DCE Application Development Guide—Introduction and Style Guide*, Revision 1.1, Prentice Hall, Englewood Cliffs, New Jersey.
- Otte, Randy, Paul Patrick, and Mark Roy, 1996, *Understanding CORBA, The Common Object Request Broker Architecture*, Prentice Hall PTR, Upper Saddle River, NJ.

Reports and Documents

- Central Imagery Office, 12 April 1996, *United States Imagery System 2000 (Target 1) Concept of Operations (CONOPS)*, CIO-2065, Washington, D.C.
- _____, 12 April 1996, *USIS Architecture Migration Plan (UAMP) Final Report*, Washington, D.C.
- _____, 12 April 1996, *UAMP Technology Assessment Report*, Washington, D.C.
- _____, 12 April 1996, *USIS Management Plan*, Washington, D.C.
- _____, 7 May 1996, *Accelerated Architecture Acquisition Initiative (A3I) Requirements Document (ARD)*, CIO-2054, Revision 2, Washington, D.C.
- _____, 5 September 1996, *CIIF Reference Model (RM)*, Version 1.6, CIO-2061, Washington, D.C.
- _____, 1996, *Functional Imagery Manipulation Requirements for the Global Command and Control System (GCCS)*, Version 1.0, Draft White Paper, Washington, D.C.
- Defense Information Systems Agency Joint Interoperability and Engineering Organization, June 28, 1996, *DII COE Programmers Manual v2.0*, U.S. Department of Defense, Washington, D.C.

_____, June 28, 1996, *DII COE Programmers Manual v2.0*, U.S. Department of Defense, Washington, D.C.

_____, June 28, 1996, *DII COE Programmers Reference Manual v2.0*, U.S. Department of Defense, Washington, D.C.

_____, June 28, 1996, *DII COE API Reference Guide v2.0.0.1 for HP and Solaris*, Volume 2, U.S. Department of Defense, Washington, D.C.

Director of Defense Information, 15 January 1993, “Interim Management Guidance on the Technical Architecture for Information Management”, U.S. Department of Defense, Washington, D.C.

DoDIIS Engineering Review Board, February 1, 1995, *A Comparison of the GCCS COE to the DoDIIS CSE*, draft, U.S. Department of Defense, Washington, D.C.

Grout, G.A., M. S. Hirsh, W. B. Ramsey, April 1995, *DoDIIS Migration Systems Instructions to DExAs, PMOs, and Developers*, MTR95W64, The MITRE Corporation, McLean, Virginia.

National Photographic Interpretation Center, 30 September 1996, *CIIF Security Analysis Report*, Washington, D.C.

Appendix A

GCCS and COE Software Segments (Version 3.0)

The following tables list the software in the COE segments that constitute the current release of the DII COE. The software is grouped into Kernel and non-Kernel components. A list is given for each of the current platforms. The platforms today are Sun Solaris and Hewlett Packard UX Unix systems and Windows NT. Other Unix platforms planned for the DII COE during Version 3 are Digital Unix, IBM AIX, and Silicon Graphics IRIX.

In the table of non-Kernel components, segments marked with an asterisk were not available in the initial release of Version 3.0, but were expected to be added during the first month or two.

Kernel 3.0.0.3 Functionality	SUN Solaris 2.4	SUN Solaris 2.5.1	Hewlett-Packard UX 9.07	Hewlett-Packard UX 10.10	Windows NT 3.51
Operating System Patches	101878-13 102224-06 101905-01 102277-02 101933-01 102292-02 101945-39 102319-01 101959-07 102664-01 101973-16 102680-03 102007-02 102704-02 102042-05 102711-01 102044-01 102756-01 102066-09 102769-03 102070-01 102922-03 102165-02 103070-01 102216-05 103290-02 102218-03	101878-13 102224-06 101905-01 102277-02 101933-01 102292-02 101945-39 102319-01 101959-07 102664-01 101973-16 102680-03 102007-02 102704-02 102042-05 102711-01 102044-01 102756-01 102066-09 102769-03 102070-01 102922-03 102165-02 103070-01 102216-05 103290-02 102218-03	PHCO_6780 PHNE_6013 PHKL_4269 PHSS_5499 PHKL_4334 PHSS_5695 PHKL_6050 PHSS_5696 PHNE_5399 PHSS_6249	PHCO_6780 PHNE_6013 PHKL_4269 PHSS_5499 PHKL_4334 PHSS_5695 PHKL_6050 PHSS_5696 PHNE_5399 PHSS_6249	Service Pack 3 or higher
Desktop	Common Desktop Environment (CDE) 1.0.0.3/TED 4.0	Common Desktop Environment (CDE) 1.0.0.3/TED 4.0	Common Desktop Environment (CDE) 1.0.0.3/TED 4.0	Common Desktop Environment (CDE) 1.0.0.3/TED 4.0	Inherent
Distributed Computing & Object Management	DCEC 1.0.0.1/1.1	DCE 1.0.0.1/1.1	DCEC 1.0.0.1/1.1	DCEC 1.0.0.1/1.1	N/A
Printing Services	Print Services 1.0.0.3	Print Services 1.0.0.3	Print Services 1.0.0.3	Print Services 1.0.0.3	Inherent
Runtime Tools	COEAskUser COEFindSeg COEInstaller COEInstError COEMsg COEPrompt COEPromptPasswd COEUpdateHome	COEAskUser COEFindSeg COEInstaller COEInstError COEMsg COEPrompt COEPromptPasswd COEUpdateHome	COEAskUser COEFindSeg COEInstaller COEInstError COEMsg COEPrompt COEPromptPasswd COEUpdateHome	COEAskUser COEFindSeg COEInstaller COEInstError COEMsg COEPrompt COEPromptPasswd COEUpdateHome	COEAskUser.exe COEFindSeg.exe COEInstaller.exe COEInstError.exe COEMsg.exe COEPrompt.exe COEPromptPasswd. exe
Security Management	Console Window 1.0.0.1/1.2.1.1 Deadman 1.0.0.1/1.2.1.2 Password 1.0.0.0/1.2.1.1 XDM 1.0.0.0/1.2.1.1	Console Window 1.2.1.1 Deadman 1.2.1.2 Password 1.2.1.1 XDM 1.2.1.1	Security Services (inherent to HP platform)	Security Services (inherent to HP platform)	Inherent
System Management	Security Manager 1.0	Security Manager 1.0	Security Manager 1.0	Security Manager 1.0	Inherent
Windowing	Motif 1.0.0.3/1.2.4 X Windows 1.0.0.3/X.11R5	Motif 1.0.0.3/1.2.4 X Windows 1.0.0.3/X.11R5	Motif 1.0.0.3/1.2.4 X Windows 1.0.0.3/X.11R5	Motif 1.0.0.3/1.2.4 X Windows 1.0.0.3/X.11R5	Inherent

COORDINATION DRAFT 18 November 1996

Function Area	SUN Solaris 2.4	SUN Solaris 2.5.1	HP UX 9.0.7	HP UX 10.10	Windows NT 3.51
Communications	UB Core 3.0.2.2 * Army Comm Server 1.4.2.4 Link 11/Tadil-A 3.0.2.2	UB Core 3.0.2.2 * Army Comm Server 1.4.2.4 Link 11/Tadil-A 3.0.2.2	UB Core 3.0.2.2 Link 11/Tadil-A 3.0.2.2	UB Core 3.0.2.2 Link 11/Tadil-A 3.0.2.2	N/A
Data Management Services	* Oracle 1.0.0.4/ 7.2.2.4 Sybase 1.0.0.3/ 10.0.2a Informix 1.0.0.1/7.12 * JCALS 1.0.0.0	* Oracle 7.2.2.4 * Sybase 11.0 Informix 1.0.0.1/ 7.12	Oracle 1.0.0.4/ 7.2.2.4 Sybase 1.0.0.3/ 10.0.2 JCALS 1.0.0.0	Oracle 7.2.2.4 Sybase 11.0	N/A
Distributed Computing & Object Management	DCES 1.0.0.4/1.1 * DCE DFS 1.0.0.0/1.1 * DCE Cell Manager 1.0.0.0/1.1 News Make Group 1.0.0.1 * WINDD 1.0.0.1	DCES 1.0.0.4/1.1 DCE DFS 1.0.0.0/1.1 DCE Cell Manager 1.0.0.0/1.1 News Make Group 1.0.0.1 * WINDD 1.0.0.1	DCES 1.0.0.4/1.1 News Make Group 1.0.0.1 * WINDD 1.0.0.1	DCES 1.0.0.4/1.1 News Make Group 1.0.0.1	N/A
Management Services	FTP Tool 1.0.0.1 GZIP 1.0.0.1/ 1.2.4 PERL 1.0.0.1/ 5.0.0.2 * NetMetrix 1.0.0.0/ 4.5.0 * Empire 1.0.0.1/ 1.35.0.2 * SPI 1.0.0.1/ 3.2.2 * Courtney 1.0.0.1 * Crack 1.0.0.0 * SATAN 1.0.0.0 TCP Wrappers 1.0.0.1 Tripwire 1.0.0.1/1.2 Tivoli 3.0.0.4 NewsPrint Software 1.0.0.2/2.5 NewsPrint Printer Config 1.0.0.1/2.5	FTPTool 1.0.0.1 GZIP 1.0.0.1/ 1.2.4 * PERL 1.0.0.1/ 5.0.0.2 * NetMetrix 1.0.0.0/4.5.0 * Empire 1.0.0.1/ 1.35.0.2 * SPI 1.0.0.1/ 3.2.2 * Courtney 1.0.0.1 * Crack 1.0.0.0 * SATAN 1.0.0.0 TCP Wrappers 1.0.0.1 Tripwire 1.0.0.1/1.2 Tivoli 3.0.0.4 NewsPrint Software 1.0.0.2/2.5 NewsPrint Printer Config 1.0.0.1/2.5	GZIP 1.0.0.1/ 1.2.4 PERL 1.0.0.1/5.0.0.2 * NetMetrix 1.0.0.0/4.5.0 * Empire 1.0.0.1/2.00b * STREAMS 1.0.0.0 * Crack 1.0.0.0 * SATAN 1.0.0.0 * TCP Wrappers 1.0.0.1 * Tripwire 1.0.0.1/1.2	GZIP 1.0.0.1/1.2.4 Perl 1.0.0.1/5.0.0.2 NetMetrix 1.0.0.0/4.5.0 Empire 1.0.0.1/2.0.0b Crack 1.0.0.0 SATAN 1.0.0.0 TCP Wrappers 1.0.0.1 Tripwire 1.0.0.1/1.2	N/A
Mapping, Charting, Geodesy & Imagery	JMTK 1.0.0.6	JMTK 1.0.0.6	JMTK 1.0.0.6	JMTK 1.0.0.6	N/A
Message Processing	IRCC 1.0.0.2/1.16 IRCS 1.0.0.1/2.8.21 MSVCS 1.0.0.2/NA TCL 1.0.0.2/7.4 * CMP 1.0.2.2 (File based) * CMP 1.0.1.2 (Informix based)	IRCC 1.0.0.2/1.16 IRCS 1.0.0.1/2.8.21 MSVCS 1.0.0.2/NA TCL 1.0.0.2/7.4 * CMP 1.0.2.2 (File based) * CMP 1.0.1.2 (Informix based)	* IRCC 1.0.0.2/1.16 IRCS 1.0.0.1/2.8.21 * MSVCS 1.0.0.2/NA * TCL 1.0.0.2/7.4	IRCC 1.0.0.2/1.16 IRCS 1.0.0.1/2.8.21 MSVCS 1.0.0.2/NA TCL 1.0.0.2/7.4	IRCC 1.0.0.0
Office Automation	Netscape Web Browser 2.0.0.2/2.0 Netscape News Server 1.0.0.2/2.0 NETSITE Server 1.0.0.1/1.1 WABI 1.0.0.2/2.2	Netscape Web Browser 2.0.0.2/2.0 Netscape News Server 1.0.0.2/2.0 NETSITE Server 1.0.0.1/1.1 WABI 1.0.0.2/2.2	Netscape Web Browser 2.0.0.2/2.0 Netscape News Server 1.0.0.2/2.0 NETSITE Server 1.0.0.1/1.1	Netscape Web Browser 2.0.0.2/2.0 Netscape News Server 1.0.0.2/2.0 NETSITE Server 1.0.0.1/1.1 WABI 1.0.0.2/2.2	Netscape Web Browser 1.0.0.1/2.0 Powerpoint 1.0.0.0/7.0 Word 1.0.0.0/7.0 Excel 1.0.0.0/7.0 MS Button Bar 1.0.0.0/4.2
Software Development Services: Developers' Tools	CalcSpace 1.0.0.4 CanInstall 1.0.0.6 ConvertSeg 1.0.0.7 MakeAttribs 1.0.0.7 MakeInstall 1.0.1.5 TestInstall 1.0.0.7 TestRemove 1.0.0.6 TimeStamp 1.0.0.6 VerifySeg 1.0.0.7 VerUpdate 1.0.1.5	CalcSpace 1.0.0.4 CanInstall 1.0.0.6 ConvertSeg 1.0.0.7 MakeAttribs 1.0.0.7 MakeInstall 1.0.1.5 TestInstall 1.0.0.7 TestRemove 1.0.0.6 TimeStamp 1.0.0.6 VerifySeg 1.0.0.7 VerUpdate 1.0.1.5	CalcSpace 1.0.0.4 CanInstall 1.0.0.6 ConvertSeg 1.0.0.7 MakeAttribs 1.0.0.7 MakeInstall 1.0.1.5 TestInstall 1.0.0.7 TestRemove 1.0.0.6 TimeStamp 1.0.0.6 VerifySeg 1.0.0.7 VerUpdate 1.0.1.5	CalcSpace 1.0.0.4 CanInstall 1.0.0.6 ConvertSeg 1.0.0.7 MakeAttribs 1.0.0.7 MakeInstall 1.0.1.5 TestInstall 1.0.0.7 TestRemove 1.0.0.6 TimeStamp 1.0.0.6 VerifySeg 1.0.0.7 VerUpdate 1.0.1.5	CalSpace 1.0.0.4 CanInstall 1.0.0.6 MakeInstall 1.0.1.5 TestInstall 1.0.0.7 TestRemove 1.0.0.6 TimeStamp 1.0.0.6 VerifySeg 1.0.0.7 VerUpdate 1.0.1.5

Appendix B**DII COE Distributed Computing Primer**

The following description of the distributed computing components of the DII COE is given in a “primer” made available by DISA at their web site. Under “Issues”, a list of current topics in the Distributed Computing Working Group is given. Note that CORBA appears in three of them.

Overview

The Distributed Computing component of the DII COE provides technology to support two software develop paradigms; Remote Procedure Call and Distributed Object Management. The core technologies that are being used are the Distributed Computing Environment (DCE), defined by the Open Group (previously the Open Software Foundation), and the Common Object Request Broker Architecture (CORBA), defined by the Object Management Group (OMG). Over time, the Distributed Computing component of the COE may evolve to include support for OLE/COM and other forms of distributed computing, such as JAVA/www.

Availability

Implementations of DCE and CORBA will be provided as part of the DII COE kernel. The client-side software for DCE included in version 3.0 of the COE kernel, will be available at no cost to COE users. The DCE servers (security, distributed file system, etc) are licensed separately, and typically only one set of servers per DCE cell is required. CORBA(2.0) is not yet provided in the COE kernel, but is planned to be included in version 4.0 of the COE kernel.

Products

DCE: Transarc Corporation’s implementation of DCE V1.1. To provide support for transaction processing and queuing, Transarc’s Encina suite of products (Encina TP monitor, Reliable Queuing Service, ...) has been recommended but has not been implemented yet. HAL Software’s DCE Cell Manager has been recommended for a GUI based DCE management.

CORBA: CORBA product recommendations have not been made yet, pending product evaluations against requirements.

Issues

DCE/CORBA compatibility/interoperability.

DCE/Ada bindings.

Army/Unixpros modifications to support dynamic reconfiguration and mobile applications.

CORBA requirements, migration, and Ada bindings

CORBA product recommendation

Manageability

Scaleability

Appendix C

CORBA Requirements in DII COE Distributed Computing SRS

The following is an excerpt of the July 1996 draft DII COE Distributed Computing SRS for Version 4.0 of the DII COE.

CORBA Specific Requirements

To address distributed computing needs for the object-oriented software development paradigm, the DII has adopted the Common Object Request Broker (CORBA) technology, defined by the Object Management Group (OMG). There are many reasons why CORBA was selected, most of which are beyond the scope of this SRS. The DII COE specific requirements for the use of CORBA are specified in the next sections.

Background

This section is included for information purposes only at this time, until CORBA requirements are transition are better understood. In the future, this section should be removed from this document.

Programs that are planning, designing, or using CORBA include:

- a) **New Attack Submarine NSSN program.** This program has specified CORBA for use in interfacing its subsystems over the C3I System network. The Prime contractor is Lockheed Martin Federal Systems Division. Lockheed Martin proposed using IONA. It will probably be the only ORB product used in the system. No work on object class definitions has been done yet. NSSN has many applications that will be based upon processors other than workstations such as: HP 743I VME board running HPRT and PowerPCs running VxWorks.
- b) **Theater Battle Management Core Systems.** The prime contractor for TBMCS (LORAL) has chosen IONA ORBIX as the ORB for design/implementation. TBMCS will integrate CTAPS, CIS, and WCCS under a single architecture.
- c) **DARPA Distributed Air Operations Center (DAOC) Advanced Technology Demonstration (ATD).** Logicon has selected IONA ORBIX as the commercial ORB.
- d) **DARPA/ISO - Joint Task Force ATD program** - provides collaborative tools for the CJTF and staff. Linked with theater CINCS and deployed forces. The architectural contractor, Teknowledge Federal Systems, has been supporting a two-ORB policy, using IONA ORBIX as the commercial ORB and Corbus, a GOTS ORB

developed by BBN. The system is currently moving to a second commercial ORB that has not been selected.

- e) **JFACC Program** - just getting underway, will provide a collaborative capability for the JFACC and staff that enables a continuous planning cycle for employment of air assets. The JFACC program will use the JTF ATD architecture as a starting point (described above).

COMMENT: The following requirements were received from Navy, but I don't think that schedule is within the scope of an SRS. Are the wrapper development efforts dependent upon CORBA being in the COE kernel? Or is it that the Navy would like to wrap the referenced products using the CORBA products recommended by the DCWG and so the Navy wants a product decision by the specified timeframe?

The time frame in which systems require CORBA technology vary; The Navy desires that some of its applications, including NIPS, TDBM, and ATWCS be wrapped with CORBA wrappers by November 1996. The Air Force's TBMCS program is currently using CORBA in design/development and will deploy some operational CORBA based capabilities beginning in 4QCY97.

CORBA Version

The implementation shall be compliant with version 2.0 of the CORBA specifications, as specified by the Object Management Group.

Note: There is not currently a validation and compliance testing suite, so compliance right now is not something that can easily be verified.

CORBA Interfaces

CORBA. The implementation shall provide implementations of the following adopted CORBA interfaces:

- a) ORB core
- b) IIOP
- c) Implementation Repository
- d) Interface Repository
- e) IDL compiler
- f) Static Invocation Interface
- g) Dynamic Invocation Interface
- h) Dynamic Skeleton Interface.

CORBA services. The implementation shall provide the following CORBA services as defined by the OMG:

- a) Naming
- b) Event Management
- c) Transaction
- d) Lifecycle
- e) Security
- f) Query
- g) Time

Note: Some of the CORBA services specified above have not yet been implemented by vendors, although they have been adopted by the OMG. Those that are specified for COE V4.0 are expected to be available within the needed time frame.

Future CORBA Services. In the future, the implementation shall provide the following additional CORBA services:

- a) Concurrency
- b) Relationship
- c) Licensing
- d) Persistence
- e) Trader
- f) Properties
- g) Externalization.

CORBA facilities

CORBA facilities. The implementation shall provide the following CORBA facilities as specified by the OMG: a) Compound Document Presentation and Data Interchange. This facility is based on the Opendoc specifications developed by IBM, Apple, CIL, et al.

CORBA Applications

Interface Repository Browser: The implementation shall provide a GUI-based capability for browsing the interfaces that are contained in the interface repositories of local and remote systems, as permitted by security policy.

CORBA Software Development

Inter-ORB traffic monitor/debugger: The implementation shall provide a GUI-based tool for monitoring CORBA traffic between clients and servers that can be used to assist in debugging the clients, servers, and CORBA configuration.

Templates: The implementation shall provide example client and server software templates that demonstrate typical usage of the common CORBA interfaces, for each of the supported programming languages.

Acronyms

ACINT	acoustic intelligence
ADP	automatic data processing
AIMS	Adopted Information Technology Standards
AOG	Architecture Oversight Group
API	application programming interface
CASE	computer -aided software engineering
CBI	computer-based instruction
CDE	Common Desktop Environment
CDS	Common Data Server
CIIF	Common Imagery Interoperability Facility
CIWGW	Common Imagery Interoperability Working Group
CIL	Command Imagery Library
CIO	Central Imagery Office
COE	common operating environment
COMINT	communications intelligence
COP	consistent operational picture
CORBA	Common Object Request Broker Architecture
DBMS	database management system
DCE	Distributed Computing Environment
DCOM	Distributed Common Object Model
DII	Defense Information Infrastructure
DIS	draft international standard
DISA	Defense Information Systems Agency
DMA	Defense Mapping Agency
DoD	Department of Defense

ELINT	electronic intelligence
GCCS	Global Command and Control System
GDMS	Global Data Management System
GUI	graphical user interface
I&RTS	Integration and Runtime Specification
IAS	Image Access Services
IBM	International Business Machines
IDL	interface definition language
IOP	Internet Interoperability Protocol
IPA	imagery product archive
ISMC	Imagery Standards Management Committee
ISO	International Standards Organization
ITS	imagery transformation services
JCALs	Joint Computer-aided Acquisition Logistics Support System
JDISS	Joint Deployable Intelligence Support System
JMTK	Joint Mapping Tool Kit
JTA	Joint Technical Architecture
MCG&I	mapping, charting, geodesy, and imagery
MIDB	Military Intelligence Data Base
MIG	MIDB and IPA for GCCS
NCCOSC	Naval Command, Control and Ocean Surveillance Center
NIL	National Imagery Library
NIMA	National Imagery and Mapping Agency
NISE	NCCOSC In Service Engineering Division
NITF	National Imagery Transmission Format
NPIC	National Photographic Interpretation Center
NT	New Technology (Microsoft operating system)

ODMG	Object Database Management Group
OMG	Object Management Group
ORB	object request broker
RPC	remote procedure call
SGI	Silicon Graphics, Incorporated
SIPRNET	Secret Internet Protocol Router Network
SHADE	Shared Data Environment
SRS	Software Requirements Specification
TADIL	Tactical Digital Information Link
TAFIM	Technical Architecture Framework for Information Management
TRM	technical reference model
TWG	technical working group
USIGS	United States Imagery and Geospatial System
USIS	United States Imagery System
USMTF	United States Message Text Format
WWMCCS	World-Wide Military Command and Control

